AD-A077 358 WYOMING UNIV LARAMIE DEPT OF MECHANICAL ENGINEERING F/G 16/1
RESEARCH ON TEST RANGE INSTRUMENTATION FOR MISSILES AND ROCKETS--ETC(U)
OCT 79 K M PELL, J E NYDAHL DAAG29-77-G-0090
UNCLASSIFIED UWME-DR-906-101-1 NL

1 OF 1
AD
A077358

END
DATE
FILMED
12-79
DDC

ARO 14855.2-A-E

(12)

RESEARCH ON
TEST RANGE INSTRUMENTATION FOR MISSILES AND ROCKETS

LEVEL

AD A077358

FINAL REPORT

KYNRIC M. PELL
and
JOHN E. NYDAHL

October, 1979

U.S. ARMY RESEARCH OFFICE

GRANT NO. DAAG 29-77-G-0090

UNIVERSITY OF WYOMING
LARAMIE, WYOMING

79 11 26 138

THE FINDINGS IN THIS REPORT ARE NOT TO BE CONSTRUED AS AN OFFICIAL DEPARTMENT OF THE ARMY POSITION, UNLESS SO DESIGNATED BY OTHER AUTHORIZED DOCUMENTS.

DISCLAIMER NOTICE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. (14)	3. RECIPIENT'S CATALOG NUMBER UWME-DR-9061011
4. TITLE (and Subtitle) (6) RESEARCH ON TEST RANGE INSTRUMENTATION FOR MISSILES AND ROCKETS.		5. TYPE OF REPORT & PERIOD COVERED (9) Final Report. Feb 77-Apr 79,
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) (10) Kynric M. Pell & John E. Nydahl		8. CONTRACT OR GRANT NUMBER(s) (15) DAAG 29-77-G-0090 Feb 77 - April 79
9. PERFORMING ORGANIZATION NAME AND ADDRESS University of Wyoming Department of Mechanical Engineering Laramie, WY 82071		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS (12) 60
11. CONTROLLING OFFICE NAME AND ADDRESS U. S. Army Research Office P. O. Box 12211 Research Triangle Park, NC 27709		12. REPORT DATE (11) 31 October 1979
		13. NUMBER OF PAGES 54
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

The view, opinions, and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Missiles, rockets, attitude, lasers, instrumentation

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A new concept, employing ground-based lasers, for attitude measurement of flight vehicles on test ranges is described. Two ground-based laser stations are required. Each station consists of a standard laser radar unit with the addition of a continuous-wave laser as well as transmitting and detecting optics for the cw laser. Cooperative vehicles incorporating two roof prisms, in addition to standard retroreflecting elements for tracking are required. → next page

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified 408 984

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Item 20 (Continued)

cont. The mathematical algorithms and sofware required to determine vehicle attitude from retroreflected pulse data are presented. Hardware required for application to non-spinning vehicles is described. A single laser-ground station has been installed on Range 1 of Redstone and has been used to determine the space position of missiles, rockets and aircraft. Hardware problems with the attitude subsystem have precluded verification of the attitude measurement concept.

Table of Contents

List of Figures

List of Appendixes

Introduction

The objective of this study was to investigate techniques for improving missile and rocket test range instrumentation. Specifically, a laser concept for measurement of vehicle position and attitude originally proposed by Conard and Pell (1,2,3) was to be exploited.

Description of the concept.

The concept has been described in detail elsewhere (4,5,6); however, a description will be included here for completeness.

Two ground-based laser tracking stations are required. Typically these stations incorporate a pulsed laser, transmitter, and detector optics all situated on an elevation over azimuth mount. The return signal from the vehicle must generally be enhanced through the use of conventional corner cubes, reflective tape, or paints located on the vehicle. Such devices are current technology, typfied by the Precision Automated Tracking System (PATS). In the following presentation it is assumed that the tracker will provide the space position of the vehicle. In order to determine attitude, each of the ground stations is additionally equipped with a continuous wave laser; and two roof prisms are located onboard the test vehicle. For vehicles exhibiting roll rates equal to or greater than the desired attitude data rates, the prisms are simply inlet into the surface of the round at the convenient location (e.g., dummy warhead). In the development which follows, we will assume that this is the case.

Consider the 90-deg roof prism shown in Fig. 1. For this application, the two surfaces which are shown crosshatched are silvered. Two lines emanating from the center of each silvered surface and perpendicular

to their respective surface define a plane, termed here the plane of the retroreflector. A simple ray trace shows that a ray of light incident on one of the silvered surfaces and describing a path parallel to the plane of the retroreflector is generally reflected off the second surface back to the origin. If a rolling vehicle is equipped with a 90-deg roof prism and tracked so that it is continuously illuminated by a cw laser, it is clear that, a return signal of cw radiation will be returned to the laser site each time the plane of the retroreflector passes through this ground station. The signal will be in the form of a pulse, the width of which depends on the beam divergence angle, the range, the optical quality of the roof prism, and the roll rate of the vehicle. The frequency of the pulses is directly related to the roll rate.

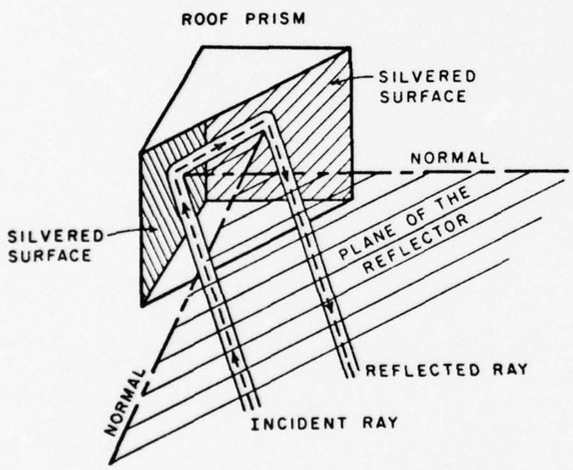

Fig. 1 Geometric optics of a roof prism.

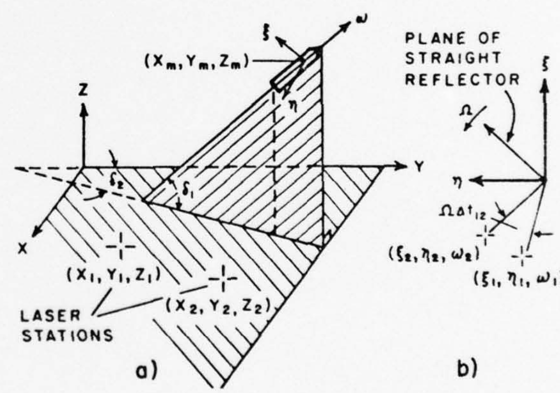

Fig. 2 Geometry for the development of the mathematical model.

A test range equipped with two tracking stations is illustrated in Fig. 2. A relationship between the return pulses received at the two stations will now be derived for a vehicle fixed in space, rotating at a constant angular rate, and equipped with a single roof prism oriented

such that the plane of the reflector contains the roll axis of the vehicle. Two coordinate systems will be utilized as shown in Fig. 2. The Earth-fixed system is defined with the origin located at the launch point; positive Z axis in the vertical upward direction; positive Y axis in the downrange direction; and the X axis in the crossrange direction to provide a right-handed system. A vehicle-centered system is defined with the origin located at the intercept of a plane bisecting the 90-deg prism and the vehicle's roll axis. The ω axis coincides with the roll axis of the vehicle, positive toward the nose; η perpendicular to ω and parallel to the XY plane, positive toward the positive X direction; and ξ perpendicular to ω and η to form a right-hand orthogonal system. The components of the position vectors of the ith ground station and the vehicle in the Earth-fixed system are (X_i, Y_i, Z_i) and (X_m, Y_m, Z_m), respectively. Using the transformation matrix between the two systems, the components of the ith ground station in the vehicle-centered system are

$$\eta=\cos(\delta_2)\,(X_i-X_m)\,-\sin(\delta_2)\,(Y_i-Y_m) \tag{1a}$$

$$\omega_i=\sin(\delta_2)\cos(\delta_1)(X_i-X_m)+\cos(\delta_2)\cos(\delta_1)(Y_i-Y_m)\,+\sin\,(\delta_1\,)\,(Z_i-Z_m) \tag{1b}$$

$$\xi_i=-\sin(\delta_2)\sin(\delta_1)(X_i-X_m)-\cos(\delta_2)\sin(\delta_1)(Y_i-Y_m)$$
$$+\cos(\delta_1)(Z_i-Z_m) \tag{1c}$$

where δ_1 and δ_2 represent the geometric pitch and yaw, respectively, defined as in Fig. 2a. Referring to Fig. 2b, it may be seen that the

time interval between pulses returned to the i and j ground station is

$$\Delta t_{ij} = \frac{1}{\Omega}\left[\arctan\left(\frac{\xi_i}{\eta_i}\right) - \arctan\left(\frac{\xi_j}{\eta_j}\right)\right] \tag{2}$$

Note that substitution for the ξ_i and η_i using Eq. (1) yields an equation in terms of relative position of the stations and the vehicle, and the geometric pitch and yaw. A third ground stations could be used to provide a similar relationship yielding two equations and two unknowns (δ_i, δ_2) where it is assumed that the relative positions are obtained from the laser tracker, and the roll rate is inferred from pulses returned to a single station. Unfortunately, simultaneous solution of these two equations is relatively insensitive to yaw variation for reasonable third station locations and suffers the disadvantage of requiring three ground stations.

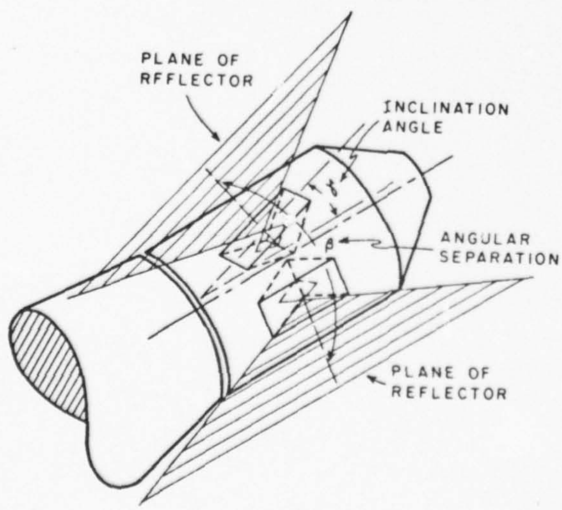

Fig. 3 Vehicle equipped with two prisms.

Consider now the addition of a second roof prism to the vehicle, as indicated in Fig. 3, with an angular separation of β relative to the first reflector and a skew of γ. The geometry of the prisms is shown in more detail in Fig. 4. Assume that the t_1 the plane of the straight retroreflector passes through the ith ground station. As the vehicle continues to roll the plane of the second reflector eventually passes through the same ground station at t_2. It is apparent from the figure that the vehicle must roll a distance $\beta + \theta$ so that the time interval between the two pulse receptions is $\Delta t_{ii} = (1/\Omega)\ (\beta + \theta_i)$.

Figure 4 also indicates that $\tan \sigma_i = R/B_i$ and $\tan \gamma = C_i/B_i$. Therefore, $\theta_i = \arcsin (\tan\gamma)/(\tan\sigma_i)$. By noting that

$$\tan \sigma_i = (\eta_i^2 + \xi_i^2)^{\frac{1}{2}}/\omega_i$$

one obtains the result

$$\Delta t_{ii} = \frac{1}{\Omega}\ \beta + \arcsin\left[\left(\frac{\omega_i \tan\gamma}{(\eta_i^2 + \xi_i^2)^{\frac{1}{2}}}\right)\right] \tag{3}$$

which can be expressed in terms of δ_1 and δ_2 using Eqs. (1).

This equation is very similar to the Yawsonde formulation because both approaches involve the relationship of planes, fixed with respect to the vehicle, relative to a remote point. The Yawsonde approach differs from the concept presented here in as much as it involves on-board detectors and telemetry. Equations (2) and (3) may be solved numerically for δ_1 and δ_2 assuming once again that Ω, $X_i - X_m$, $Y_i - Y_m$, $Z_i - Z_m$, Δt_{12} are known.

Unfortunately, a closed form solution of these two equations could not be found. The problem was reformulated to achieve a closed form solution as explained in the next section.

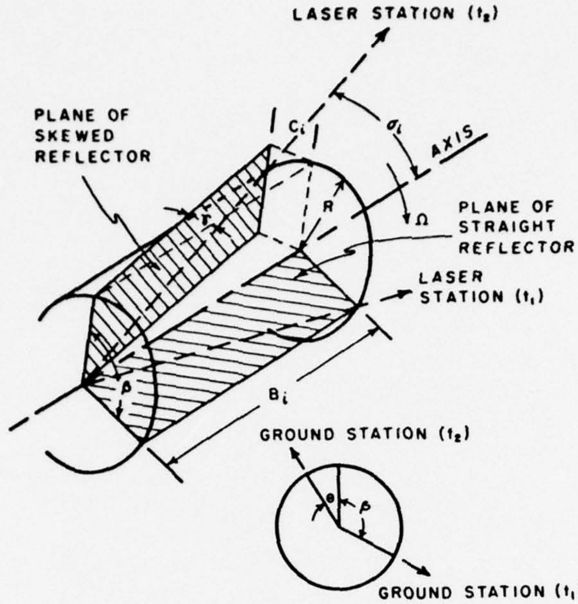

Fig. 4 Retroreflection planes referenced to a laser ground station at two different times.

Reformulation of the mathematical description

An alternate expression for the time interval for retroreflection from one retroreflector to the two ground stations can be developed.

Two ground stations are located relative to an earth fixed coordinate system by vectors $\vec{r}_i$. The position of a missile rotating with a constant angular velocity ω about a roll axis $\hat{\varepsilon}_\omega$ is given by vector $\vec{r}_m$. From these vectors, we form unit vectors $\hat{\varepsilon}_{ri}$ describing the direction from the missile to the ground stations:

$$\hat{\varepsilon}_{ri} \equiv \frac{\vec{r}_i - \vec{r}_m}{|\vec{r}_i - \vec{r}_i|} \quad i = 1,2 \tag{4}$$

In this equation the carat (^) denotes a vector of unit magnitude and $|\vec{r}_i - \vec{r}_m|$ denotes the magnitude of the vector difference between $\vec{r}_i$ and $\vec{r}_m$.

A retroreflection plane may be defined as a plane within which a light signal transmitted to the missile is returned to the place of emission. A unit normal which defines the plane of retroreflection currently passing through ground station (i) may be defined as $\hat{\varepsilon}_{\tau i}$, where $\hat{\varepsilon}_{\tau i}$ is physically interpretable as a unit vector lying along the intersection of two mirror surfaces placed 90° apart.

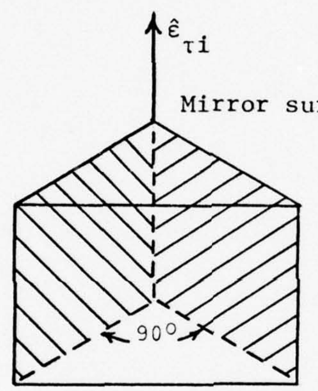

Fig. 5 Definition of $\hat{\varepsilon}_{\tau i}$

We define a skewness angle α as the angle between $\hat{\varepsilon}_{\tau i}$ and a normal to the roll axis ($\hat{\varepsilon}_{ni}$) when $\hat{\varepsilon}_{\tau i}$, $\hat{\varepsilon}_{\omega}$, and $\hat{\varepsilon}_{ni}$ are all coplanar and which allows for mounting the mirrors skewed relative to the roll axis.

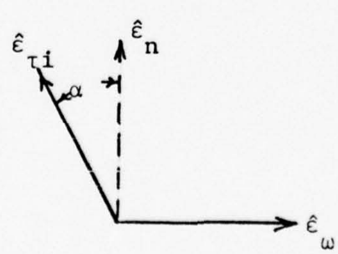

Fig 6. Definition of α

The angle between $\hat{\varepsilon}_{ri}$ and $\hat{\varepsilon}_{\omega}$ is the tracking aspect angle σ_i.

$$\hat{\varepsilon}_{ri} \times \hat{\varepsilon}_{\omega} = \sin \sigma_i \, \hat{\varepsilon}_{ni} \tag{5}$$

The angle between the two ground stations as taken from the missile is defined as μ.

$$\hat{\varepsilon}_{r1} \cdot \hat{\varepsilon}_{r2} = \cos \mu_{12} \tag{6}$$

The retroreflection plane is assumed to pass through ground station 1 at time t_1. It passes through station 2 at time $t_2 = t_1 + t_{12}$. Between these two times, the retroreflection plane has rotated through an angle $\omega\Delta t_{12}$ where ω is the constant roll rate of the missile.

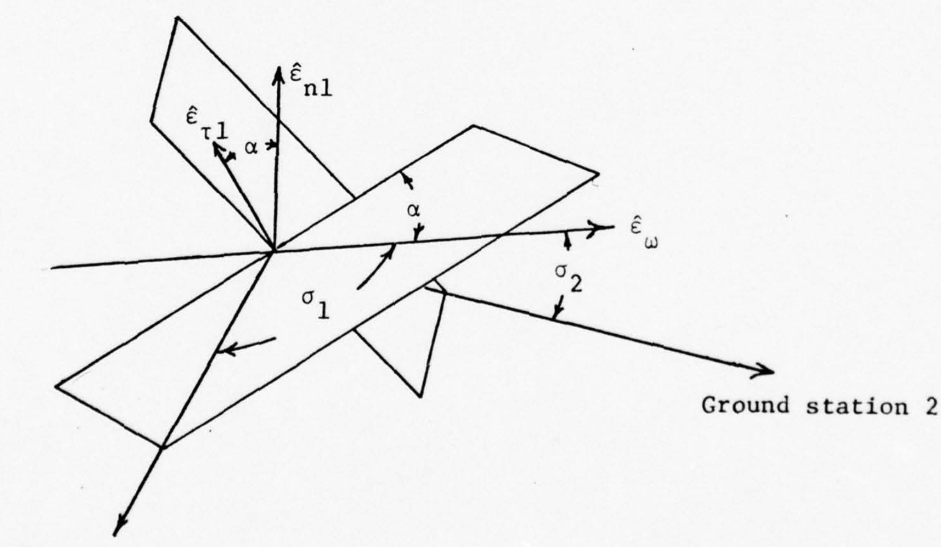

Fig. 7 Geometric relationship

In order to determine the angle of rotation, we must define a unit vector $\hat{\varepsilon}_{Ai}$ that lies in the plane of retroreflection with station i, but is perpendicular to the roll axis $\hat{\varepsilon}_{\omega}$. It is necessary that $\hat{\varepsilon}_{Ai}$ be a linear combination of vector $\hat{\varepsilon}_{ri}$ and $\hat{\varepsilon}_{\omega}$. This $\hat{\varepsilon}_{Ai}$ is found using the form

$$\hat{\varepsilon}_{Ai} = \frac{\hat{\varepsilon}_{\omega} \times \hat{\varepsilon}_{\tau}}{|\hat{\varepsilon}_{\omega} \times \hat{\varepsilon}_{\tau}|} \tag{7}$$

Now that all angles and vectors needed to find the aspect angles σ_i have been found, we may define $\hat{\varepsilon}_{\tau i}$.

$$\hat{\varepsilon}_{\tau i} = -\sin\alpha\ \hat{\varepsilon}_{\omega} + \frac{\cos\alpha}{\sin\sigma_i} \hat{\varepsilon}_{ri} \times \hat{\varepsilon}_{\omega} \tag{8}$$

Using equations (7) and (8), and the fact that

$$|\hat{\varepsilon}_{\omega} \times \hat{\varepsilon}_{\tau i}| = |\sin(90 + \alpha)| = \cos\alpha \tag{9}$$

we arrive at

$$\hat{\varepsilon}_{Ai} = \frac{1}{\sin\sigma_i} [\hat{\varepsilon}_{ri} - \cos\sigma_i\ \hat{\varepsilon}_{\omega}] \tag{10}$$

As previously mentioned, the angle of roll that the missile has experience between retroreflecting station 1 and 2 is $\omega\Delta t_{12}$. The vectors $\hat{\varepsilon}_{A1}$ and $\hat{\varepsilon}_{A2}$ must also have undergone this same angular change so that

$$\hat{\varepsilon}_{A1} \cdot \hat{\varepsilon}_{A2} = \cos(\omega\Delta t_{12})$$

Using equation (10) twice, we see that

$$\begin{aligned}
\cos(\omega\Delta t_{12}) &= \hat{\varepsilon}_{A1} \cdot \hat{\varepsilon}_{A2} \\
&= \frac{1}{\sin\sigma_1 \sin\sigma_2} [\hat{\varepsilon}_{r1} - \cos\sigma_1\ \hat{\varepsilon}_{\omega}] \cdot [\hat{\varepsilon}_{r2} - \cos\sigma_2\ \hat{\varepsilon}_{\omega}] \\
&= \frac{1}{\sin\sigma_1 \sin\sigma_2} [\hat{\varepsilon}_{r1} \cdot \hat{\varepsilon}_{r2} - \cos\sigma_1\ \hat{\varepsilon}_{\omega} \cdot \hat{\varepsilon}_{r2} \\
&\quad -\cos\sigma_2\ \hat{\varepsilon}_{r1} \cdot \hat{\varepsilon}_{\omega} + \cos\sigma_1 \cos\sigma_2\ \hat{\varepsilon}_{\omega} \cdot \hat{\varepsilon}_{\omega}]
\end{aligned}$$

$$= \frac{1}{\sin \sigma_1 \sin \sigma_2} [\cos \mu_{12} - \cos \sigma_1 \cos \sigma_2$$

$$-\cos \sigma_2 \cos \sigma_1 + \cos \sigma_1 \cos \sigma_2]$$

$$= \frac{1}{\sin \sigma_1 \sin \sigma_2} [\cos \mu_{12} - \cos \sigma_1 \cos \sigma_2] \quad (12)$$

With equation (3) written in the form

$$t_{i1} = \frac{1}{\Omega} [\beta + \arcsin (\frac{\tan\gamma}{\tan\sigma_1})] \quad (13)$$

Equations (12) and (13) can be solved simultaneously for σ_1 and σ_2 in closed form. The two aspect angles can be used to determine the orientation of the longitudinal axis of the vehicle thereby providing vehicle attitude. A solution alogorithm has been implemented in program ASP11 which is documented in Appendix I. Input data for ASP11 includes ground station locations, missile position, roll rate and measured time intervals.

In order to obtain an accurate missile position in range coordinates from the data tape generated by LAMPAMS a second program was developed.

Vehicle position determination

Data tapes generated by the LAMPAMS are in binary form, 9 track, 800 bpi. The data arrangement on the tape is shown in Fig. 8 . In order to generate accurate position information the azimuth, elevation and range recorded on the tape in binary form are:

1. Transformed to based 10.
2. Corrected for bias.
3. Smoothed with a running 10 point least squares polynomial.
4. Transformed to range coordinates.

KEY - PATS DATA (First sixteen bytes in each sample)

Time: (BCD) H, hours
M, minutes
S, seconds
ms, milliseconds

Encoders: (natural binary, LSB = $360°/2^{18}$) AZ, azimuth angle
EL, elevation angle

Range: (Binary, LSB = 1.0 foot)

Status bits:
- LIM, limit
- RA, range
- RR, recording rate
  - 00 = 10 samples per second
  - 01 = 20 samples per second
  - 10 = 50 samples per second
  - 11 = 100 samples per second
- LP, transmitter power (unused)
- ROV, range overflow
- SP, signal present
- EM, event marker switches
- AE, azimuth tracking error
- RM, remote mode (unused)
- RDV, remote data valid flag (unused)
- LO, laser on
- CM, computer track mode
- AT, autotrack mode
- TQ, track quality switch
- EC, event count
- SI, system ID
- ST, station ID

The "EPOCH OFFSET" is encoded as follows:

0 0 0 0 = -10 milliseconds
0 0 0 1 = -9
0 0 1 0 = -8
0 0 1 1 = -7
0 1 0 0 = -6
1 0 0 0 = -5
1 0 0 1 = -4
1 0 1 0 = -3
1 0 1 1 = -2
1 1 0 0 = -1

Time of Attitude Pulse = EPOCH TIME (time from "PATS" data in same sample) +EPOCH OFFSET + PULSE POSITION

	MSB							LSB
TIME 1	20H	10H	8H	4H	2H	1H	40M	20M
TIME 2	10M	8M	4M	2M	1M	40S	20S	10S
TIME 3	8S	4S	2S	1S	.8S	.4S	.2S	.1S
TIME 4	80ms	40ms	20ms	10ms	8ms	4ms	2ms	1ms
AZENC 1	AZ16------AZ9							
AZENC 2	AZ8------AZ1							
ELENC 1	EL16------EL9							
ELENC 2	EL8------EL1							
RANGE 1	RA16------RA9							
RANGE 2	RA8------RA1							
AEL MSB'S	AZ18	AZ17	EL18	EL17	RA18	RA17		LIM
STATUS 1	RR1	RR0	LP1	LP0	ROV	SP	EM1	EM2
STATUS 2	AE5	AE4	AE3	AE2	AE1	EE5	EE4	EE3
STATUS 3	EE2	EE1	RM	RDV	LO	CM	AT	TQ
EVENT	EC7	EC6	EC5	EC4	EC3	EC2	EC1	EC0
IDENT	SI2	SI1	SI0	ST4	ST3	ST2	ST1	ST0
	ATTITUDE DATA FRAME 1 (4 BYTES)							
	ATTITUDE DATA FRAME 2 (4 BYTES)							
	⋮							
	ATTITUDE DATA FRAME 10 (4 BYTES)							

First byte of sample, i.e., the first character of the tape.

56 bytes per sample

First byte of next sample, et.

There are 32 samples per record or 1792 bytes per record.

ATTITUDE DATA (Last 40 bytes in each sample -4 bytes per frame, 10 frames per sample)

NOTE: Attitude data frames of all zeros should be ignored.

One frame of valid attitude data is as shown below:

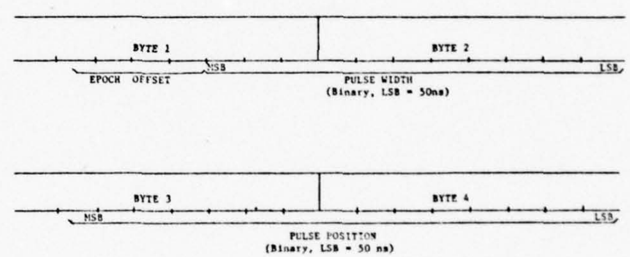

Fig. 8 LAMPAMS magnetic tape format

A program (LTPOS) has been written to accomplish these tasks and a listing is included as Appendix II. This program was initially developed to run on the central computer at the U.S. Army Missile Command (CDC 6600) in anticipation that this is where the data analysis would be done. Subsequently, the program was adapted to run on the Hewlett Packard 1000 System, which was installed on the test range at Redstone Arsenal.

An additional program to determine tracker bias based on measurements of surveyed points equipped with retroreflectors was acquired from Yuma proving ground and adapted to run on the central computer at Redstone. This program has been partially converted to run on the Hewlett Packard system; however, this has not been completed. Because of the length of the program overlay techniques are required and considerable additional software development will be required to accomplish this task.

Application to non-rolling vehicles

The approach evolved for testing non-rolling or slowly rolling vehicles involves spinning a portion of the dummy warhead up to relatively high rates prior to launch and simply allowing it to spin-down during flight. In order to demonstrate the approach, twelve 2.75 inch rocket fuses were reworked as shown in Figure 9 and provided to MICOM for use in flight tests. Design of a pre-spin device was coordinated with MICOM personnel and a unit was fabricated at MICOM.

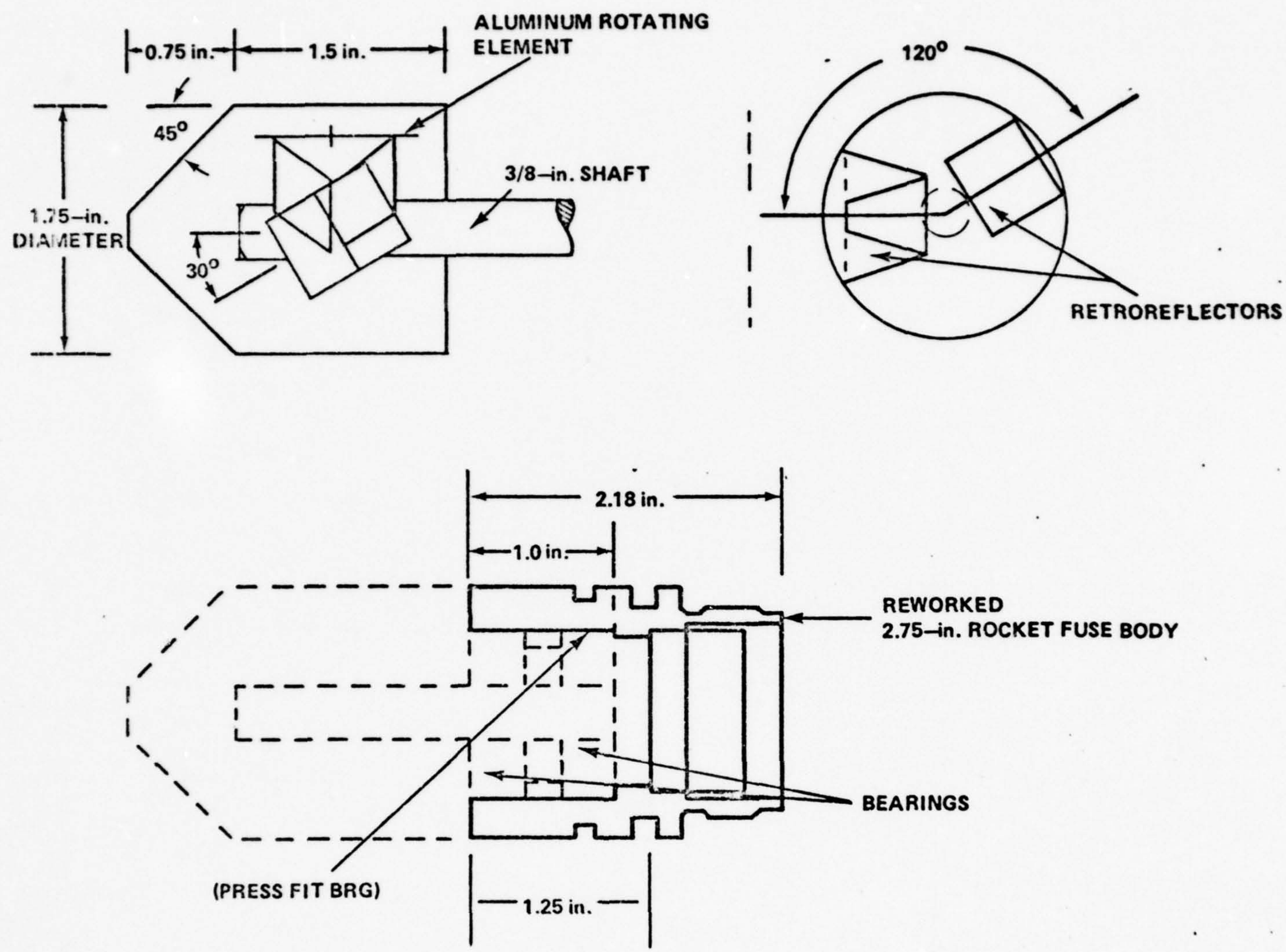

Fig. 9 Reworked 2.75 inch fuse assembly

Status of the LAMPAMS

The LAMPAMS system was installed on Range 1 of Redstone Arsenal in the spring of 1978. Since that time the tracker portion of the system has become the primary instrument for vehicle position determination. The program LTPOS originally provided to MICOM has served as the basis for data reduction. It has been extensively revised by MICOM personnel to include graphical output.

Unfortunately, the CW laser and signal conditioning hardware associated with the attitude subsystem have been subject to repeated failures and as of September 1979 no attitude data has been generated.

Since both the pulsed laser used for tracking and the CW laser on the attitude system operate at the same frequency the data must be separated on the basis of pulse width in software. In addition, the potential for missing a return pulse from the attitude subsystem is higher than that associated with the tracking system. The software package required to determine appropriate time intervals for use in ASP11 has not been completed because of these uncertainties and the lack of data to resolve the areas in question.

Conclusions and Recommendations

Determination of vehicle attitude using the concept described requires two laser ground stations. Presently, only one station is available. The concept can however, be verified using a single ground station. Every effort should be made to get the attitude subsystem of LAMPAMS operational so that the required software can be completed. The efficacy of the concept could then be verified. This should be completed prior to acquisition of a second laser ground station.

References

1. "A Laser System for Determination of Rocket Attitude and Roll Rate," with Mark J. Russell, John E. Nydahl, and William Lindberg. Final Report, grant DAHCO4074-0063/Department of Mechanical Engineering Report UWME-DR-4061051, July 1974.

2. "Measurement of Missile Position and Attitude by Lasers," with R.G. Conard and J.E. Nydahl. MICOM Report TR-RD-76-7, July 1975.

3. "Techniques for Remote Sensing of Missile Position and Attitude," MICOM Report TR-RD-76-11, October 1975.

4. "A Laser System to Measure Rocket Attitude in Flight," with John E. Nydahl et al., Journal of Spacecraft and Rockets, Vol. 15, No. 4, July-August 1978

5. "Preliminary Test Results on a Laser System for Measurement of Rocket Attitude During Flight Test," with J.E. Nydahl et al., paper 78-17, Space Science Meeting, AIAA, Huntsville, AL January 1978

6. "Attitude Determination of Missiles and Rockets Using a Laser System," with J. E. Nydahl, R. G. Conard and C.R. Cooke, Proceedings of the 9th Annual Electo Optics Conference, Anaheim, CA, October 1977

Appendix I

ASP11

Program to determine attitude

of vehicle

(Hewlett Packard System 1000 version)

THE MAIN PROGRAM

The main program exists to call the proper input subroutine for reduction of input values to pitch and yaw angles. It also sets the input/output device numbers, the maximum allowable error, the constant π, and the number of degrees per radian.

Currently the main programcalls only one other program segment, SUBROUTINE INPUT.

SUBROUTINE INPUT

The subroutine provides an interactive method of operating the rest of the program. Sixteen commands give versatility and ease to changing any or all of the parameters involved in a pitch and yaw determination. The subroutine prompts the user for required input of the various parameters. As an added check, the values fed in after a prompt are displayed to the user. If any doubt remains as to what variables the program is using, a command is available to display the current values of all variables.

The first time through, the user is prompted to input all the necessary variables for a computation of pitch and yaw. After all variables are input, a prompt for a command is printed. At this point fifteen possible commands are available:

RUN - will determine the pitch and yaw using the current variables

A RETURN, with nothing entered - does the same as RUN

DISPLAY - displays a list of all current variables

? - provides a list of commands that are available and what they do

GSI - will change the coordinates of Ground Station One

GS2 - changes Ground Station Two coordinates

MISSILE - changes missile coordinates

DT11 - changes the time difference for the two mirrors to retroreflect through ground station one

DT12 - changes the time difference for one mirror to retroreflect through ground station one, and then ground station two

BETA - changes the radial separation between the two mirrors on the missile

SK1 - changes the skewness of mirror set one

SK2 - changes the skewness of mirror set two

OMEGA - changes the roll rate of the missile

RESTART - starts an entirely new case where all variables must be reentered

STOP - halts the execution of the program when you are finished

Any command may be input after the prompt "COMMAND", and in any order.

nega, the roll rate, may be input in

RAD - radians per second

RPS - revolutions per second

RPM - revolutions per minute

DPS - degrees per second

e program converts the input value into radians per second.

When INPUT encounters a RUN command, or a blank command, subroutine SOLVE called to determine the pitch and yaw angles.

SUBROUTINE SOLVE

SOLVE is the organizer for the solution of pitch and yaw angles. It calls TUP to calculate the geometric relationships between the ground stations and e missile. Next, SIG11 is called to determine the first aspect angle. SIG12 subsequently called to find what the other aspect angle is. ESUBW is then lled to find the correct roll axis.

SOLVE always returns to INPUT.

SUBROUTINE SETUP

SETUP determines $\vec{R1}$, $\vec{R2}$, $\hat{e}_{R1}$, and $\hat{e}_{R2}$. Unit vector $\hat{e}_{R1}$ is parallel to $\vec{R1}$. The magnitues R1 and R2 are determined, as is COSMU which is the cosine of the angle between $\vec{R1}$ and $\vec{R2}$.

SUBROUTINE SIG11

Using the time necessary for both sets of missile mirrors to retroreflect through ground station one (DT11), and the yawsonde equation, the first aspect angle SIGMA1 is determined.

SUBROUTINE SIG12

The time required for one mirror to retroreflect through ground station one and then ground station two (DT12), when used in

$$\cos(\Omega \Delta t_{12}) = \frac{1}{\sin\sigma_1 \sin\sigma_2} [\cos\mu - \cos\sigma_1 \cos\sigma_2] \tag{1}$$

where: Ω = OMEGA, the missile roll rate in radians per second

Δt_{12} = DT12

$\cos\mu$ = COSMU

$\cos\sigma_1$ = cos(SIGMA1)

$\sin\sigma_1$ = sin(SIGMA1)

determines two possible SIGMA2 aspect angles. SIG11 provides SIGMA1 (σ_1), SETUP provides COSMUμ(cos), so that σ_2 may be determined explicitly.

Two roots become possible for σ_2. These are SIG2A and SIG2B. To further complicate matters, we only accept angles that are $0 \leq \sigma_2 \leq \pi$. When the arctangent function is used, a negative argument may represent either $-\sigma_2$ or $\pi/2 - \sigma_2$.

These difficulties were circumvented by:

1) replacing a $\sigma_2 < 0$ by $\pi/2 - \sigma_2$, hence making σ_2 positive

2) testing the resultant σ_{2A} and σ_{2B} to see if equation (1) is satisfied.

When the subroutine returns to SOLVE, ISOL contains the integer number of solutions which satisfy (1). If ISOL = 0, no solution is possible, and SOL1 = SOL2 = 0, an arbitrary default value. If ISOL = 1, the correct solution is in SOL1. If ISOL = 2 the two solutions are in SOL1 and SOL2.

SUBROUTINE ESUBW

When two aspect angles are given, the roll axis, $\hat{\varepsilon}_w$ (ESUBW) may be determined by the solution of:

$$\begin{aligned} \hat{\varepsilon}_{R1} \cdot \hat{\varepsilon}_w &= \cos(\sigma_1) \\ \hat{\varepsilon}_{R2} \cdot \hat{\varepsilon}_w &= \cos(\sigma_2) \\ \hat{\varepsilon}_w \cdot \hat{\varepsilon}_w &= 1 \end{aligned} \tag{2}$$

Equations (2) are nonlinear, and therefore not easily solved. The method used in this program determines the $\hat{i}$ component of $\hat{\varepsilon}_w$, or ε_{wx}. This reduces (2) to:

$$\begin{pmatrix} \varepsilon_{R1S} & \varepsilon_{R1Y} & \varepsilon_{R1Z} \\ \varepsilon_{R2X} & \varepsilon_{R2Y} & \varepsilon_{R2Z} \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} \varepsilon_{wx} \\ \varepsilon_{xy} \\ \varepsilon_{wz} \end{pmatrix} = \begin{pmatrix} \cos(\sigma_1) \\ \cos(\sigma_2) \\ \varepsilon_{wx} \end{pmatrix} \tag{3}$$

Equation (3) is linear, and easily solved with a SIMultaneous eQuation solver, SIMQ. With the roll axis known, pitch and yaw are easily determined by:

$$\text{YAW} = \text{ARCTAN}\ (\varepsilon_{wx}/\varepsilon_{xy})$$

$$\text{PITCH} = \text{ARCTAN}[\varepsilon_{wx}/(\varepsilon_{wx}^2 + \varepsilon_{wy}^2)^{1/2}]$$

The three cases which are impossible to solve using this formulation

are

Case I $\varepsilon_{R1Y} = \varepsilon_{R2Y} = 0$

Case II $\varepsilon_{R1Z} = \varepsilon_{R2Z} = 0$

Case III $\hat{\varepsilon}_{R1} = K\ \hat{\varepsilon}_{R2}$ where K is a real number

Case III physically corresponds to the missile being located between the two ground stations along a vecotr connecting the ground stations.

Cases I and II reduce equation (3) to an overdeterminant set which may be solved for ε_{wx} and ε_{wz}, and ε_{wx} and ε_{wy} respectively, as a system of two equations in two unknowns. The remaining component may be found by

$$\varepsilon_{wx}{}^2 + \varepsilon_{wy}{}^2 + \varepsilon_{wz}{}^2 = 1 \qquad (5)$$

The main difficulty would be the determination of ε_{wx}. This is done in subroutine ESUBWX.

It is important to note that, given two aspect angles, there are two possible roll axes. To determine the correct roll axis, subroutine TIME is called. When TIME is given a roll axis, it determines the two time differences DELT11 and DELT12 from previous work. If the two computed time differences are within the allowable error limit, then the roll axis is valid, otherwise it is not.

SUBROUTINE ESUBWX

ESUBWX explicitly determines ε_{wx} from equation (2). There are generally two solutions ε_{wx}. These reside in ROOT1 and ROOT2. Occasionally the two roots are redundant, then ISOL, the number of solutions is one. If $\hat{\varepsilon}_R 1$ and $\hat{\varepsilon}_{R2}$ are linearly dependent, then ISOL = 0 as no solution is possible.

In the solution of ε_{wx}, a quadratic equation of the form

$$(1 + B^2 + D^2)\varepsilon^2_{wx} + 2(AB + CD)\varepsilon_{xw} + (A^2 + C_2 - 1) = 0 \qquad (6)$$

is recurrent. The values of A, B, C and D vary depending on the vectors $\hat{\epsilon}_{R1}$, $\hat{\epsilon}_{R2}$, and the values of $\cos(\sigma_1)$ and $\cos(\sigma_2)$. Subroutine QUADRA solves quadratic equation (6) given A, B, C, and D.

SUBROUTINE QUADRA

QUADRA solves equation (6) given A, B, C and D. If the radical resulting is less than zero, ISOL = 0, and no solution is possible because the two roots would be complex. This corresponds to two non-intersecting aspect angle cones. If ISOL = 1, one solution is the same as the other. This corresponds to two tangent cones. Typically ISOL = 2, and two solutions for ϵ_{wx}, ROOT1 and ROOT2, exist.

SUBROUTINE SIMQ

Given an N by N matrix [A], and a vector [B] (N x 1), SIMQ solves the matrix equation

$$[A]\ [X] = [B] \tag{7}$$

for [X], a vector that is N x 1. During the course of computation, [A] and [B] are destroyed.

A possible alteration to speed computation would be to use Cramer's rule for the solution of [X] instead of the current Gauss elimination procedure.

SUBROUTINE TIME

TIME calculates DELT11 and DELT12 when given a roll axis and all geometric constants.

Pitch and yaw are determined from equations (4). An alternate coordinate system is defined as

A possible alteration to speed computation would be to use Cramer's

rule for the solution of [X] instead of the current Gauss elimination procedure.

SUBROUTINE TIME

TIME calculates DELT11 and DELT12 when given a roll axis and all geometric constants.

Pitch and yaw are determined from equations (4). An alternate coordinate system is defined as

$$\eta_i = \cos(\text{YAW})\ RX_i - \sin(\text{YAW})RY_i \tag{8}$$

$$\omega_i = \sin(\text{YAW})\ \cos(\text{PITCH})RX_i + \cos(\text{PITCH})RY_i + \sin(\text{PITCH})RZ_i$$

$$\varepsilon_i = \sin(\text{YAW})\ \sin(\text{PITCH})RX_i - \cos(\text{YAW})\ \sin(\text{PITCH})RY_i + \cos(\text{PITCH})RZ_i$$

times are found by

$$\text{DELT11} = \frac{1}{\text{OMEGA}}\ \{\text{BETA} + \text{ARCSIN}\ [\omega_1\ \text{TAN(SKEW1)}/(\varepsilon_1{}^2 + \eta_1{}^2)^{\frac{1}{2}}]\} \tag{9}$$

$$\text{DELT12} = \frac{1}{\text{OMEGA}}\ \ \text{ARCTAN}\ (\varepsilon_1/\eta_1) - \text{ARCTAN}\ (\varepsilon_2/\eta_2)]$$

where OMEGA is the foll rate of the missile in radians per second

BETA is the angular separation of the mirrors in radians

SKEW1 is the skewness of mirror one

If both of the DELT times are within the allowable error tolerance of the initial times given, then the pitch and yaw values are output.

```
FTN4,L
      PROGRAM ASPII
      DOUBLE PRECISION X1,Y1,Z1,X2,Y2,Z2,XM,YM,ZM,SKEW1,
     1 SKEW2,BETA,RX1,RY1,RZ1,R1,RX2,RY2,RZ2,R2,COSMU,XR1,
     2 YR1,ZR1,XR2,YR2,ZR2,ANSWER,PI,RADIAN,
     3 ERROR,EWX,EWY,EWZ
      INTEGER OUTPUT
      COMMON X1,Y1,Z1,X2,Y2,Z2,XM,YM,ZM,SKEW1,
     1 SKEW2,BETA,RX1,RY1,RZ1,R1,RX2,RY2,RZ2,R2,COSMU,XR1,
     2 YR1,ZR1,XR2,YR2,ZR2,ANSWER(4,6),PI,RADIAN,
     3 ERROR,EWX,EWY,EWZ,NUMSOL,NREAD,OUTPUT
      DIMENSION IPRAM(5)
      CALL RMPAR(IPRAM)
      NUMSOL=0
      NREAD=IPRAM(1)
      OUTPUT=IPRAM(2)
      PI=4.D0*DATAN(1.D0)
      RADIAN=180.D0/PI
      ERROR=1.D-5
      CALL INPUT
      STOP
      END
```

```
      SUBROUTINE INPUT
      DOUBLE PRECISION DT11,DT12
      DOUBLE PRECISION BETAIN,SKEW1I,SKEW2I
      DOUBLE PRECISION OMEGA,OMEGIN,PITCH,YAW
      DOUBLE PRECISION X1,Y1,Z1,X2,Y2,Z2,XM,YM,ZM,SKEW1,
     1 SKEW2,BETA,RX1,RY1,RZ1,R1,RX2,RY2,RZ2,R2,COSMU,XR1,
     2 YR1,ZR1,XR2,YR2,ZR2,ANSWER,PI,RADIAN,
     3 ERROR,EWX,EWY,EWZ
      INTEGER OUTPUT
      COMMON X1,Y1,Z1,X2,Y2,Z2,XM,YM,ZM,SKEW1,
     1 SKEW2,BETA,RX1,RY1,RZ1,R1,RX2,RY2,RZ2,R2,COSMU,XR1,
     2 YR1,ZR1,XR2,YR2,ZR2,ANSWER(4,6),PI,RADIAN,
     3 ERROR,EWX,EWY,EWZ,NUMSOL,NREAD,OUTPUT
      INTEGER ICONT(32)
      DATA ICONT/2HGS,2H1 ,2HGS,2H2 ,2HMI,2HSS,2HDT,2H11,2HDT,
     1 2H12,2HBE,2HTA,2HSK,2H1 ,2HRU,2HN ,2H  ,2H  ,2HST,2HOP,
     2 2HDI,2HSP,2HSK,2H2 ,2HOM,2HEG,2HRE,2HST,2H? ,2H  ,2HSO,
     3 2HLU/
      DATA IRAD1,IRPS2,IRPM2,IDPS1/2HRA,2HS ,2HM ,2HDP/
    5 KFLAG=0
   10 WRITE(OUTPUT,100)
  100 FORMAT("  INPUT GROUND STATION 1 COORDS: X,Y,Z")
      READ(NREAD,*)X1,Y1,Z1
      WRITE(OUTPUT,101)X1,Y1,Z1
  101 FORMAT(1P3(2X,D20.15))
      IF(KFLAG.NE.0) GOTO 300
   20 WRITE(OUTPUT,102)
  102 FORMAT("  INPUT GROUND STATION 2 COORDS: X,Y,Z")
      READ(NREAD,*)X2,Y2,Z2
      WRITE(OUTPUT,101)X2,Y2,Z2
      IF(KFLAG.NE.0) GO TO 300
   30 WRITE(OUTPUT,103)
  103 FORMAT("  INPUT MISSILE COORDS: X,Y,Z")
      READ(NREAD,*)XM,YM,ZM
      WRITE(OUTPUT,101)XM,YM,ZM
      IF(KFLAG.NE.0) GO TO 300
   40 WRITE(OUTPUT,104)
  104 FORMAT("  INPUT DELTA T 11, IN SECONDS")
      READ(NREAD,*)DT11
      WRITE(OUTPUT,105)DT11
      IF(KFLAG.NE.0) GO TO 300
  105 FORMAT(2X,1PD23.15)
   50 WRITE(OUTPUT,106)
  106 FORMAT("  INPUT DELTA T 12, IN SECONDS")
      READ(NREAD,*)DT12
      WRITE(OUTPUT,105) DT12
      IF(KFLAG.NE.0) GO TO 300
   60 WRITE(OUTPUT,107)
  107 FORMAT("  INPUT THE RADIAL SEPARATION OF MIRRORS IN "
     1 "DEGREES")
      READ(NREAD,*)BETAIN
      BETA=BETAIN/RADIAN
      WRITE(OUTPUT,105)BETAIN
      IF(KFLAG.NE.0) GO TO 300
   70 WRITE(OUTPUT,108)
  108 FORMAT("  INPUT THE SKEW ANGLE OF MIRROR 1, IN DEGREES")
      READ(NREAD,*)SKEW1I
      SKEW1=SKEW1I/RADIAN
      WRITE(OUTPUT,105)SKEW1I
  110 FORMAT(/"  WHEN YOU WISH TO CHANGE SOMETHING, OR RUN"/
     1 "  THE PROG., THEN FOLLOW THE LIST BELOW"/
     2"  GS1=GROUND STATION ONE COORDINATES"/
     2"  GS2=GROUND STATION TWO COORDS.  "/"  MISSILE=MISSILE COORDS."/
     4"  SK1=SKEW OF MIRROR 1"/"  SK2=SKEW OF MIRROR 2"/"  RUN=COMPUTE"
     5/"  NO ENTRY, JUST A RETURN, WILL EXEC THE PROG"/"  STOP="
     6"HALT OF EXECUTION"/"  BETA=RADIAL SEPARATION OF THE MIRRORS"/
     7"  DISPLAY=CURRENT VALUES OF ALL VARIABLES"/)
      IF(KFLAG.NE.0) GO TO 300
   71 WRITE(OUTPUT,109)
```

```
  109 FORMAT("  INPUT THE SKEW ANGLE OF MIRROR 2, IN DEGREES")
      READ(NREAD,*)SKEW2I
      SKEW2=SKEW2I/RADIAN
      WRITE(OUTPUT,105)SKEW2I
      IF(KFLAG.NE.0)GO TO 300
   72 WRITE(OUTPUT,115)
      READ(NREAD,*)OMEGIN
      READ(NREAD,112)IUNIT1,IUNIT2
  114 FORMAT(2X,D23.15,2A2)
  115 FORMAT("  INPUT THE ROLL RATE, SKIP A LINE,"/
     1 "  AND THEN RAD FOR RADIANS PER SECOND"/
     2 "          RPS FOR REVOLUTIONS PER SECOND"/
     3 "          RPM FOR REVOLUTIONS PER MINUTE"/
     4 "          DPS FOR DEGREES PER SECOND")
      OMEGA=0.D0
      IF(IUNIT1.EQ.IRAD1)OMEGA=OMEGIN
      IF(IUNIT2.EQ.IRPS2)OMEGA=2.D0*PI*OMEGIN
      IF(IUNIT2.EQ.IRPM2)OMEGA=120.D0*PI*OMEGIN
      IF(IUNIT1.EQ.IDPS1)OMEGA=OMEGIN/RADIAN
      IF(OMEGA.EQ.0.D0) GO TO 91
      WRITE(OUTPUT,116)OMEGIN,IUNIT1,IUNIT2,OMEGA
  116 FORMAT("  YOUR  "1PD16.10,2X,2A2"CONVERTS TO"F16.10" RADIANS PER"
     1 " SECOND")
      KFLAG=1
      IF(KFLAG.EQ.0) WRITE(OUTPUT,110)
  111 FORMAT("  COMMAND")
  300 WRITE(OUTPUT,111)
      READ(NREAD,112)ICOM1,ICOM2
  112 FORMAT(2A2)
      IFLAG=0
      DO 301 I=1,16
      IF(ICONT(2*I-1).EQ.ICOM1.AND.ICONT(2*I).EQ.ICOM2)IFLAG=I
  301 CONTINUE
      IF(IFLAG.EQ.0) WRITE(OUTPUT,113)
      IF(IFLAG.EQ.0) GO TO 300
  113 FORMAT("  EH?")
      GO TO(10,20,30,40,50,60,70,80,80,90,200,71,72,5,73,74),IFLAG
   91 WRITE(OUTPUT,113)
      GO TO 300
   73 WRITE(OUTPUT,110)
      GO TO 300
   74 WRITE(OUTPUT,117)
      DO 119 INT=1,NUMSOL
  119 WRITE(OUTPUT,118)INT,ANSWER(INT,1),ANSWER(INT,2),ANSWER(INT,3)
     1 ,ANSWER(INT,4),ANSWER(INT,5),ANSWER(INT,6)
  117 FORMAT(/"  SOLUTION    PITCH      YAW          TIME11    TIME12"
     1 "      ERR11      ERR12"/)
  118 FORMAT(2X,I5,2F10.5,2X,2F10.6,2D10.4)
      GO TO 300
   90 RETURN
   80 CALL SOLVE(PITCH,YAW,OMEGA,DT11,DT12)
      GO TO 300
  200 WRITE(OUTPUT,201)X1,Y1,Z1,X2,Y2,Z2,XM,YM,ZM,BETAIN,SKEW1I,
     1 SKEW2I,DT11,DT12,OMEGA
  201 FORMAT("  STATION ONE:"2(D16.9",")D16.9/"  STATION TWO:"
     12(D16.9",")D16.9/"  MISSILE:"2(D16.9",")D16.9/"  BETA="D16.9" DEG"
     2 ,"REES"/"  SKEW1="D16.9" DEGREES"/"  SKEW2="D16.9" DEGREES",
     3 /"  DELTA T 11="D16.9" SECONDS"/"  DELTA T 12=",
     4 D16.9" SECONDS"/"  OMEGA="D16.9" RADIANS PER SECOND"/)
      GO TO 300
      END
```

```
C SIMQ SOLVES THE MATRIX EQUATION A TIMES X EQUALS B
C FOR THE VECTOR X
C GIVEN ARE THE N BY N MATRIX A, AND THE VECTOR B.
C WHEN KS=0, A VALID SOLUTION IS GIVEN
C WHEN KS=1, THE MATRIX A IS SINGULAR
      SUBROUTINE SIMQ(A,B,KS)
      DOUBLE PRECISION A(3,3),B(3),DETA,DETB,DETC,DET
      KS=0
      DET=A(1,1)*(A(2,2)*A(3,3)-A(2,3)*A(3,2))-
     1    A(1,2)*(A(2,1)*A(3,3)-A(2,3)*A(3,1))+
     2    A(1,3)*(A(2,1)*A(3,2)-A(3,1)*A(2,2))
      IF(DET.EQ.0.D0) KS=1
      IF(DET.EQ.0.D0)RETURN
      DETA=B(1)*(A(2,2)*A(3,3)-A(2,3)*A(3,2))-
     1    A(1,2)*(B(2)*A(3,3)-A(2,3)*B(3))+
     2    A(1,3)*(B(2)*A(3,2)-B(3)*A(2,2))
      DETB=A(1,1)*(B(2)*A(3,3)-A(2,3)*B(3))-
     1    B(1)*(A(2,1)*A(3,3)-A(2,3)*A(3,1))+
     2    A(1,3)*(A(2,1)*B(3)-A(3,1)*B(2))
      DETC=A(1,1)*(A(2,2)*B(3)-B(2)*A(3,2))-
     1    A(1,2)*(A(2,1)*B(3)-B(2)*A(3,1))+
     2    B(1)*(A(2,1)*A(3,2)-A(3,1)*A(2,2))
      B(1)=DETA/DET
      B(2)=DETB/DET
      B(3)=DETC/DET
      RETURN
      END
```

```
C E SUB W IS THE ROLL AXIS OF THE MISSILE.  GIVEN THE TWO ASPECT ANGLES
C THIS ROUTINE WILL DETERMINE THE ROLL AXIS UNIT VECTOR
C AS WELL AS THE PITCH AND YAW FOR THE PARTICULAR
C CASE IN QUESTION
      SUBROUTINE ESUBW(SIGMA1,SIGMA2,OMEGA,DT11,DT12)
      DOUBLE PRECISION DT11,DT12
      DOUBLE PRECISION SIGMA1,SIGMA2
      DOUBLE PRECISION OMEGA,ROOT1,ROOT2
      DOUBLE PRECISION DELT11,DELT12,COSS1,COSS2,DET
      DOUBLE PRECISION X1,Y1,Z1,X2,Y2,Z2,XM,YM,ZM,SKEW1,
     1 SKEW2,BETA,RX1,RY1,RZ1,R1,RX2,RY2,RZ2,R2,COSMU,XR1,
     2 YR1,ZR1,XR2,YR2,ZR2,ANSWER,PI,RADIAN,
     3 ERROR,EWX,EWY,EWZ
      INTEGER OUTPUT
      COMMON X1,Y1,Z1,X2,Y2,Z2,XM,YM,ZM,SKEW1,
     1 SKEW2,BETA,RX1,RY1,RZ1,R1,RX2,RY2,RZ2,R2,COSMU,XR1,
     2 YR1,ZR1,XR2,YR2,ZR2,ANSWER(4,6),PI,RADIAN,
     3 ERROR,EWX,EWY,EWZ,NUMSOL,NREAD,OUTPUT
      DOUBLE PRECISION AA(3,3),A(3,3),B(3),C(3)
      IF(YR1.EQ.YR2.AND.YR1.EQ.0.D0) GO TO 90
      IF(ZR1.EQ.ZR2.AND.ZR1.EQ.0.D0) GO TO 95
      AA(1,1)=XR1
      AA(2,1)=XR2
      AA(3,1)=1.D0
      AA(1,2)=YR1
      AA(2,2)=YR2
      AA(3,2)=0.D0
      AA(1,3)=ZR1
      AA(2,3)=ZR2
      AA(3,3)=0.D0
      CALL ESWX(SIGMA1,SIGMA2,ROOT1,ROOT2,ISOL,DT11,DT12)
      IF(ISOL.GT.0) GO TO 10
      WRITE(OUTPUT,20)
      RETURN
   20 FORMAT("  **NO SOLUTION AVAILABLE FOR THE ROLL AXIS**"/
     1 "  **ESWX DID NOT RETURN A VALUE FOR EWX**"/)
   10 COSS1=DCOS(SIGMA1)
      COSS2=DCOS(SIGMA2)
      C(1)=COSS1
      C(2)=COSS2
      C(3)=ROOT1
      DO 1000 I=1,3
      B(I)=C(I)
      DO 1000 J=1,3
 1000 A(I,J)=AA(I,J)
      CALL SIMQ(AA,C,KS)
      IF(KS.EQ.0) GO TO 30
      WRITE(OUTPUT,40)
   40 FORMAT("  ***ERROR IN SIMQ, THE MATRIX FOR FINDING THE"/
     1 "  ROLL AXIS IS SINGULAR*****")
      RETURN
   30 EWX=C(1)
      EWY=C(2)
      EWZ=C(3)
      CALL TIME(DELT11,DELT12,OMEGA,DT11,DT12)
      IF(DABS(DELT11-DT11).GT.ERROR) GO TO 50
      IF(DABS(DELT12-DT12).GT.ERROR) GO TO 50
      RETURN
   50 IF(ISOL.EQ.1) RETURN
      B(3)=ROOT2
      CALL SIMQ(A,B,KS)
      IF(KS.EQ.0) GO TO 60
      WRITE(OUTPUT,40)
      RETURN
   60 EWX=B(1)
      EWY=B(2)
      EWZ=B(3)
      CALL TIME(DELT11,DELT12,OMEGA,DT11,DT12)
      IF(DABS(DELT11-DT11).GT.ERROR) GO TO 70
      IF(DABS(DELT12-DT12).GT.ERROR) GO TO 70
      RETURN
   70 WRITE(OUTPUT,80)
   80 FORMAT("  **ESUBW**THERE IS NO SUITABLE ROLL AXIS**")
      RETURN
```

```
C YR1=YR2=0.D0
   90  DET=XR1*ZR2-XR2*ZR1
       IF(DET.NE.0.D0) GO TO 91
       WRITE(OUTPUT,93)
   93  FORMAT("  **R1 AND R2 ARE NOT LINEARLY INDEPENDENT**"/
     1 "  *****NO SOLUTION POSSIBLE*****")
       RETURN
   91  COSS1=DCOS(SIGMA1)
       COSS2=DCOS(SIGMA2)
       EWX=(COSS1*ZR2-ZR1*COSS2)/DET
       EWZ=(XR1*COSS2-XR2*COSS1)/DET
       EWY=DSQRT(1.D0-EWX**2-EWZ**2)
       RETURN
   95  COSS1=DCOS(SIGMA1)
       COSS2=DCOS(SIGMA2)
       DET=XR1*YR2-YR1*XR2
       IF(DET.NE.0.D0) GO TO 96
       WRITE(OUTPUT,93)
       RETURN
   96  EWX=(COSS1*YR2-YR1*COSS2)/DET
       EWY=(XR1*COSS2-XR2*COSS1)/DET
       EWZ=DSQRT(1.D0-EWX**2-EWY**2)
       RETURN
       END
```

```
C SIGMA 1 IS DETERMINED BY THE YAWSONDE EQUATION IN THIS ROUTINE
      SUBROUTINE SIG11(OMEGA,SIGMA1,DT11)
      DOUBLE PRECISION DT11,DTAN
      DOUBLE PRECISION OMEGA,A,ALPHA,SIGMA1
      DOUBLE PRECISION X1,Y1,Z1,X2,Y2,Z2,XM,YM,ZM,SKEW1,
     1 SKEW2,BETA,RX1,RY1,RZ1,R1,RX2,RY2,RZ2,R2,COSMU,XR1,
     2 YR1,ZR1,XR2,YR2,ZR2,ANSWER,PI,RADIAN,
     3 ERROR,EWX,EWY,EWZ
      INTEGER OUTPUT
      COMMON X1,Y1,Z1,X2,Y2,Z2,XM,YM,ZM,SKEW1,
     1 SKEW2,BETA,RX1,RY1,RZ1,R1,RX2,RY2,RZ2,R2,COSMU,XR1,
     2 YR1,ZR1,XR2,YR2,ZR2,ANSWER(4,6),PI,RADIAN,
     3 ERROR,EWX,EWY,EWZ,NUMSOL,NREAD,OUTPUT
      A=DATAN(SKEW2)/DATAN(SKEW1)
      ALPHA=OMEGA*DT11-BETA
      SIGMA1=DATAN((DTAN(SKEW1)*DSQRT(1.D0+A*A+2.D0*A*DCOS(ALPHA)
     1 ))/DSIN(ALPHA))
      IF(SIGMA1.LT.0.D0)SIGMA1=SIGMA1+PI
      RETURN
      END
```

```
C THE FOLLOWING CALCULATES THE SOLUTION OF SIGMA 2 FROM
C THE ONE PRISM, TWO GROUND STATION FORMULATION
      SUBROUTINE SIG12(OMEGA,SIGMA1,DT12,SOL1,SOL2,ISOL)
      DOUBLE PRECISION DT12,DIFF2A,DIFF2B,DSIG2A,DSIG2B
      DOUBLE PRECISION OMEGA,SIGMA1,DTAN
      DOUBLE PRECISION A,SOL1,SOL2,SIG2A,SIG2B,B,RAD2,RAD,COSMEG
      DOUBLE PRECISION COSS1,COSS2A,COSS2B,SINS1,SINS2A,SINS2B
      DOUBLE PRECISION X1,Y1,Z1,X2,Y2,Z2,XM,YM,ZM,SKEW1,
     1 SKEW2,BETA,RX1,RY1,RZ1,R1,RX2,RY2,RZ2,R2,COSMU,XR1,
     2 YR1,ZR1,XR2,YR2,ZR2,ANSWER,PI,RADIAN,
     3 ERROR,EWX,EWY,EWZ
      INTEGER OUTPUT
      COMMON X1,Y1,Z1,X2,Y2,Z2,XM,YM,ZM,SKEW1,
     1 SKEW2,BETA,RX1,RY1,RZ1,R1,RX2,RY2,RZ2,R2,COSMU,XR1,
     2 YR1,ZR1,XR2,YR2,ZR2,ANSWER(4,6),PI,RADIAN,
     3 ERROR,EWX,EWY,EWZ,NUMSOL,NREAD,OUTPUT
      A=DTAN(SIGMA1)*DCOS(OMEGA*DT12)
      ISOL=0
      SOL1=0.D0
      SOL2=0.D0
      B=COSMU/DCOS(SIGMA1)
      RAD2=A*A-B*B+1.D0
      IF(RAD2.GT.0.D0) GO TO 10
      WRITE(OUTPUT,100)
  100 FORMAT("  *****ERROR IN THE SIGMA 12 SUBROUTINE*****")
      SIG2A=0.D0
      RETURN
   10 RAD=DSQRT(RAD2)
      COSMEG=DCOS(OMEGA*DT12)
      SIG2A=DATAN((A*B-RAD)/(B+A*RAD))
      SIG2B=DATAN((A*B+RAD)/(B-A*RAD))
      IF(SIG2A.LT.0.D0)SIG2A=SIG2A+PI
      IF(SIG2B.LT.0.D0)SIG2B=SIG2B+PI
      COSS1=DCOS(SIGMA1)
      COSS2A=DCOS(SIG2A)
      SINS1=DSIN(SIGMA1)
      SINS2A=DSIN(SIG2A)
      COSS2B=DCOS(SIG2B)
      SINS2B=DSIN(SIG2B)
      DIFF2A=DABS(COSMU-COSS1*COSS2A-COSMEG*SINS1*SINS2A)
      DIFF2B=DABS(COSMU-COSS1*COSS2B-COSMEG*SINS1*SINS2B)
      DSIG2A=SIG2A*RADIAN
      DSIG2B=SIG2B*RADIAN
      IF(DIFF2A.GT.ERROR)GO TO 20
C SIG2A IS A GOOD ROOT
      IF(DIFF2B.LT.ERROR) GO TO 15
C SIG2A GOOD,SIG2B BAD
      SOL1=SIG2A
      ISOL=1
      RETURN
C BOTH ROOTS ARE BAD
   12 WRITE(OUTPUT,110)
  110 FORMAT("  **THE TWO SOLUTIONS OF SIGMA2 ARE INCORRECT**")
      WRITE(OUTPUT,120)DSIG2A,DIFF2A,DSIG2B,DIFF2B
  120 FORMAT("  THE FIRST SOLUTION OF SIGMA 2 IS "D19.13/
     1 "  WITH AN ERROR OF "D19.13/
     2 "  THE SECOND SOLUTION IS "D19.13/
     3 "  WITH AN ERROR OF "D19.13)
      WRITE(OUTPUT,999) SIG2A,SIG2B,COSS1.SINS1,COSS2A,SINS2A,COSMU
     1 ,COSMEG,RAD,SIGMA1,OMEGA,DT12,A,B,COSS2B,SINS2B
  999 FORMAT("  SIG2A="1PD23.16"  SIG2B="D23.16"  COSS1="D23.16/
     1 "  SINS1="D23.16"  COSS2A="D23.16"  SINS2A="D23.16/
     2 "COSMU="D23.16"  COSMEG="D23.16"  RAD="D23.16"  SIGMA1="D23.16/
     3 "  OMEGA="D23.16"  DT12="D2.16"  A="D23.16"  B="D23.16
     4 /"  COSS2A="D23.16"  SINS2B="D23.16/)
      RETURN
C BOTH ROOTS ARE GOOD
   15 ISOL=2
      SOL1=SIG2A
      SOL2=SIG2B
      RETURN
C SIG2A IS BAD, SIG2B IS GOOD
   20 IF(DIFF2B.GT.ERROR) GO TO 12
      SOL1=SIG2B
      ISOL=1
      RETURN
      END
```

```
C SUBROUTINE SETUP DETERMINES ALL OF THE VARIABLES IN THE
C COMMON BLOCK
      SUBROUTINE SETUP
      DOUBLE PRECISION X1,Y1,Z1,X2,Y2,Z2,XM,YM,ZM,SKEW1,
     1 SKEW2,BETA,RX1,RY1,RZ1,R1,RX2,RY2,RZ2,R2,COSMU,XR1,
     2 YR1,ZR1,XR2,YR2,ZR2,ANSWER,PI,RADIAN,
     3 ERROR,EWX,EWY,EWZ
      INTEGER OUTPUT
      COMMON X1,Y1,Z1,X2,Y2,Z2,XM,YM,ZM,SKEW1,
     1 SKEW2,BETA,RX1,RY1,RZ1,R1,RX2,RY2,RZ2,R2,COSMU,XR1,
     2 YR1,ZR1,XR2,YR2,ZR2,ANSWER(4,6),PI,RADIAN,
     3 ERROR,EWX,EWY,EWZ,NUMSOL,NREAD,OUTPUT
      RX1=X1-XM
      RY1=Y1-YM
      RZ1=Z1-ZM
      R1=DSQRT(RX1**2+RY1**2+RZ1**2)
      RX2=X2-XM
      RY2=Y2-YM
      RZ2=Z2-ZM
      R2=DSQRT(RX2**2+RY2**2+RZ2**2)
      COSMU=(RX1*RX2+RY1*RY2+RZ1*RZ2)/(R1*R2)
C MAKE ALL THE PLANAR COMPONENTS NORMALISED
      XR1=RX1/R1
      YR1=RY1/R1
      ZR1=RZ1/R1
      XR2=RX2/R2
      YR2=RY2/R2
      ZR2=RZ2/R2
C ZERO OUT THE ANSWER ARRAY, AND NUMSOL(NUMBER OF SOLUTIONS)
      DO 10 I=1,4
      DO 10 J=1,6
   10 ANSWER(I,J)=0.D0
      NUMSOL=0
      RETURN
      END
```

```
C SOLVE IS THE MAIN ORGANIZER FOR THE EXPLICIT SOLUTION
C OF PITCH AND YAW ANGLES GIVEN DELTA T 11 AND DELTA T 12
C THIS ROUTINE IS RESPONSIBLE FOR THE CORRECT LINKING OF ALL
C OTHER SUBROUTINES WITH THE EXCEPTION OF INPUT
      SUBROUTINE SOLVE(PITCH,YAW,OMEGA,DT11,DT12)
      DOUBLE PRECISION DT11,DT12
      DOUBLE PRECISION OMEGA,SIGMA1
      DOUBLE PRECISION SIG2A,SIG2B
      DOUBLE PRECISION PITCH,YAW
      DOUBLE PRECISION X1,Y1,Z1,X2,Y2,Z2,XM,YM,ZM,SKEW1,
     1 SKEW2,BETA,RX1,RY1,RZ1,R1,RX2,RY2,RZ2,R2,COSMU,XR1,
     2 YR1,ZR1,XR2,YR2,ZR2,ANSWER,PI,RADIAN,
     3 ERROR,EWX,EWY,EWZ
      INTEGER OUTPUT
      COMMON X1,Y1,Z1,X2,Y2,Z2,XM,YM,ZM,SKEW1,
     1 SKEW2,BETA,RX1,RY1,RZ1,R1,RX2,RY2,RZ2,R2,COSMU,XR1,
     2 YR1,ZR1,XR2,YR2,ZR2,ANSWER(4,6),PI,RADIAN,
     3 ERROR,EWX,EWY,EWZ,NUMSOL,NREAD,OUTPUT
      DIMENSION ITIME(5)
C     CALL BEGTIM
      CALL EXEC(11,ITIME,IYEAR)
      ISTART=1000*ITIME(2)+10*ITIME(1)
      CALL SETUP
      CALL SIG11(OMEGA,SIGMA1,DT11)
      CALL SIG12(OMEGA,SIGMA1,DT12,SIG2A,SIG2B,IGOOD)
      IF(IGOOD.EQ.0)RETURN
      CALL ESUBW(SIGMA1,SIG2A,OMEGA,DT11,DT12)
      IF(IGOOD.EQ.2)CALL ESUBW(SIGMA1,SIG2B,OMEGA,DT11,DT12)
C     ITIME=0
C     CALL MARKTIM(ITIME)
      CALL EXEC(11,ITIME,IYEAR)
      IFINIS=1000*ITIME(2)+10*ITIME(1)
      JTIME=IFINIS-ISTART
      WRITE(OUTPUT,1001)JTIME
 1001 FORMAT(2X,43("*")/"  THE TIME REQUIRED FOR THIS COMPUTATION WAS"
     1 /I19" MILLISECONDS"/2X43("*")/)
      RETURN
      END
```

```
C QUAD FINDS THE QUADRATIC ROOTS TO A SPECIALIZED EQUATION
      SUBROUTINE QUAD(A,B,C,D,ROOT1,ROOT2,ISOL,OUTPUT)
      DOUBLE PRECISION ROOT1,ROOT2,A,B,C,D,AQUAD,BQUAD,CQUAD,RADSQ,RAD
      INTEGER OUTPUT
      ROOT1=0.D0
      ROOT2=0.D0
      ISOL=0
      AQUAD=1.D0+B*B+D*D
C NOTE THAT AQUAD IS ALWAYS GREATER THAN ONE
      BQUAD=2.D0*(A*B+C*D)
      CQUAD=A*A+C*C-1.D0
      RADSQ=BQUAD**2-4.D0*AQUAD*CQUAD
      IF(RADSQ.GE.0.D0) GO TO 10
      WRITE(OUTPUT,100)
C THERE IS AN ERROR APPARENT
  100 FORMAT(/"  ***ERROR IN QUADRATIC SOLUTION***"/
     1 "    CANNOT TAKE THE SQUARE ROOT OF"/
     2 "      A NEGATIVE NUMBER"/)
      RETURN
   10 RAD=DSQRT(RADSQ)
      ISOL=2
      IF(RAD.EQ.0.D0)ISOL=1
C THE TWO SOLUTIONS ARE IDENTICAL
      ROOT1=-(BQUAD+RAD)/(2.D0*AQUAD)
      ROOT2=(-BQUAD+RAD)/(2.D0*AQUAD)
      RETURN
      END
```

```
C ESWX SOLVES THE FOLLOWING SET OF EQUATIONS FOR EWX
C     ER1 DOT EW = COS(SIGMA1)
C     ER2 DOT EW = COS(SIGMA2)
C      EW DOT EW = 1.D0
C ROOT1 IS THE FIRST SOLUTION FOR EWX
C ROOT2 IS THE SECOND SOLUTION
C ISOL IS THE NUMBER OF SOLUTIONS FOR THE SET
      SUBROUTINE ESWX(SIGMA1,SIGMA2,ROOT1,ROOT2,ISOL,DT11,DT12)
      DOUBLE PRECISION SIGMA1,SIGMA2,DT11,DT12
      DOUBLE PRECISION X1,Y1,Z1,X2,Y2,Z2,XM,YM,ZM,SKEW1,
     1 SKEW2,BETA,RX1,RY1,RZ1,R1,RX2,RY2,RZ2,R2,COSMU,XR1,
     2 YR1,ZR1,XR2,YR2,ZR2,ANSWER,PI,RADIAN,
     3 ERROR,EWX,EWY,EWZ
      INTEGER OUTPUT
      COMMON X1,Y1,Z1,X2,Y2,Z2,XM,YM,ZM,SKEW1,
     1 SKEW2,BETA,RX1,RY1,RZ1,R1,RX2,RY2,RZ2,R2,COSMU,XR1,
     2 YR1,ZR1,XR2,YR2,ZR2,ANSWER(4,6),PI,RADIAN,
     3 ERROR,EWX,EWY,EWZ,NUMSOL,NREAD,OUTPUT
      DOUBLE PRECISION A,B,C,D,ROOT1,ROOT2,COSS1,COSS2
      DOUBLE PRECISION DENOM,DENNEW
C SET UP THE ERROR DEFAULTS
      ROOT1=0.D0
      ROOT2=0.D0
      ISOL=0
C GET THE DIRECTION COSINES FOR THE ASPECT ANGLES
      COSS1=DCOS(SIGMA1)
      COSS2=DCOS(SIGMA2)
C NOW GO TO THE SOLUTION FOR THE NINE ZILLION CASES
      IF((ZR1*ZR2).NE.0.D0)GO TO 110
      IF(ZR1.EQ.0.D0) GO TO 10
      IF(ZR2.EQ.0.D0) GO TO 5
      WRITE(OUTPUT,1000)
 1000 FORMAT(" ***ERROR--SOME TYPE OF LOGIC MISTAKE***")
      RETURN
    5 IF(YR2.EQ.0.D0)GO TO 6
      A=COSS2/YR2
      B=-XR2/YR2
      C=(COSS1-A*YR1)/ZR1
      D=-(XR1+B*YR1)/ZR1
      CALL QUAD(A,B,C,D,ROOT1,ROOT2,ISOL,OUTPUT)
      RETURN
    6 IF(XR2.EQ.0.D0) GO TO 7
      ISOL=1
      ROOT1=COSS2/XR2
      RETURN
    7 WRITE(OUTPUT,1010)
 1010 FORMAT(" **VECTOR ER2 IS THE ZERO VECTOR, NO SOLUTION**")
      RETURN
C ZR1=0.D0
   10 IF(ZR2.EQ.0.D0)GO TO 11
C ZR2.NE.0.D0
      IF(YR1.EQ.0.D0) GO TO 12
      A=COSS1/YR1
      B=-XR1/YR1
      C=(COSS1-YR2*A)/ZR2
      D=-(B*YR2+XR2)/ZR2
      CALL QUAD(A,B,C,D,ROOT1,ROOT2,ISOL,OUTPUT)
      RETURN
C ZR1=YR1=0.D0
   12 IF(XR1.EQ.0.D0) GO TO 13
      ISOL=1
      ROOT1=COSS1/XR1
      RETURN
   13 WRITE(OUTPUT.1020)
 1020 FORMAT(" **VECTOR ER1 IS THE ZERO VECTOR, NO SOLUTION**")
      RETURN
```

```
C ZR1=0
   11 DENOM=XR1*YR2-XR2*YR1
      IF(DENOM.EQ.0.D0) GO TO 14
      ISOL=1
      ROOT1=YR2*COSS1-COSS2*YR1
      RETURN
   14 WRITE(OUTPUT,1040)
C ZR1.NE.0.D0, AND ZR2.NE.0.D0--
C THE MOST COMMON CASE
  110 DENOM=ZR2*YR1-YR2*ZR1
      IF(DENOM.EQ.0.D0) GO TO 120
      A=(ZR2*COSS1-ZR1*COSS2)/DENOM
      B=(XR2*ZR1-ZR2*XR1)/DENOM
      C=(COSS1-YR1*A)/ZR1
      D=-(XR1+B*YR1)/ZR1
      CALL QUAD(A,B,C,D,ROOT1,ROOT2,ISOL,OUTPUT)
      RETURN
C DENOM=0.D0
  120 DENNEW=ZR2*XR1-XR2*ZR1
      IF(DENNEW.EQ.0.D0) GO TO 130
      ISOL=1
      ROOT1=(ZR2*COSS1-ZR1*COSS2)/DENNEW
      RETURN
  130 WRITE(OUTPUT,1040)
 1040 FORMAT("  **ER1 AND ER2 ARE NOT LINEARLY INDEPENDENT, "
     1 "NO SOLUTION")
      RETURN
      END
```

```
C THIS ROUTINE CALCULATES THE TIME REQUIRED FOR DELT11
C AND DELT12 TO ROLL THROUGH THE RESPECTIVE GROUND STATIONS
      SUBROUTINE TIME(DELT11,DELT12,OMEGA,DT11,DT12)
      DOUBLE PRECISION X1,Y1,Z1,X2,Y2,Z2,XM,YM,ZM,SKEW1,
     1 SKEW2,BETA,RX1,RY1,RZ1,R1,RX2,RY2,RZ2,R2,COSMU,XR1,
     2 YR1,ZR1,XR2,YR2,ZR2,ANSWER,PI,RADIAN,
     3 ERROR,EWX,EWY,EWZ
      INTEGER OUTPUT
      COMMON X1,Y1,Z1,X2,Y2,Z2,XM,YM,ZM,SKEW1,
     1 SKEW2,BETA,RX1,RY1,RZ1,R1,RX2,RY2,RZ2,R2,COSMU,XR1,
     2 YR1,ZR1,XR2,YR2,ZR2,ANSWER(4,6),PI,RADIAN,
     3 ERROR,EWX,EWY,EWZ,NUMSOL,NREAD,OUTPUT
      DOUBLE PRECISION DT11,DT12,DTAN
      DOUBLE PRECISION OMEGA,DASIN
      DOUBLE PRECISION XYMAG,PITCH,YAW,PITOUT,YAWOUT,SINPIT
      DOUBLE PRECISION SINYAW,COSYAW,COSPIT,N1,W1,E1,N2,E2,DELT11
      DOUBLE PRECISION DELT12,ERR11,ERR12
C     DOUBLE PRECISION DEG11,DEG12
      XYMAG=DSQRT(EWX**2+EWY**2)
      PITCH=DATN2(EWZ,XYMAG)
      YAW=DATN2(EWX,EWY)
      PITOUT=PITCH*RADIAN
      YAWOUT=YAW*RADIAN
      SINYAW=DSIN(YAW)
      SINPIT=DSIN(PITCH)
      COSYAW=DCOS(YAW)
      COSPIT=DCOS(PITCH)
      N1=COSYAW*RX1-SINYAW*RY1
      W1=SINYAW*COSPIT*RX1+COSYAW*COSPIT*RY1+SINPIT*RZ1
      E1=-SINYAW*SINPIT*RX1-COSYAW*SINPIT*RY1+COSPIT*RZ1
      N2=COSYAW*RX2-SINYAW*RY2
      E2=-SINYAW*SINPIT*RX2-COSYAW*SINPIT*RY2+COSPIT*RZ2
      DELT11=(BETA+DASIN(W1*DTAN(SKEW1)/DSQRT(E1*E1+N1*N1)))/OMEGA
      DELT12=(DATN2(E1,N1)-DATN2(E2,N2))/OMEGA
C     DEG11=OMEGA*DELT11*RADIAN
C     DEG12=OMEGA*DELT12*RADIAN
      ERR11=DABS(DELT11-DT11)
      ERR12=DABS(DELT12-DT12)
      NUMSOL=NUMSOL+1
      ANSWER(NUMSOL,1)=PITOUT
      ANSWER(NUMSOL,2)=YAWOUT
      ANSWER(NUMSOL,3)=DELT11
      ANSWER(NUMSOL,4)=DELT12
      ANSWER(NUMSOL,5)=ERR11
      ANSWER(NUMSOL,6)=ERR12
      IF(DABS(PITOUT).GT.90.D0.OR.DABS(YAWOUT).GT.90.D0)RETURN
      IF(DABS(DELT12-DT12).GT.ERROR) RETURN
      WRITE(OUTPUT,82)PITOUT,YAWOUT
      WRITE(OUTPUT,999)DELT11,DELT12
  999 FORMAT("  DELT11="1PD23.16"  DELT12="D23.16)
   82 FORMAT("  USING PITCH="2X,F11.6"  DEG AND YAW="2X,F11.6"  DEG")
C     WRITE(OUTPUT,401)DEG11,DEG12
C401  FORMAT(/" DELTA T 11 ROLLS THROUGH"F20.15" DEGREES"/
C    1 " DELTA T 12 ROLLS THROUGH"F20.15" DEGREES")
      CALL ERR(DT11,DT12,OMEGA)
      RETURN
      END
```

```
C THIS IS A ROUTINE TO AUGMENT THE MINI-COMPUTER
      DOUBLE PRECISION FUNCTION DASIN(X)
      DOUBLE PRECISION X
      DASIN=DATN2(X,DSQRT(1.D0-X*X))
      RETURN
      END
C ANOTHER SUBROUTINE FOR THE EDIFICATION OF A MINI-COMPUTER
      DOUBLE PRECISION FUNCTION DTAN(X)
      DOUBLE PRECISION X
      DTAN=DSIN(X)/DCOS(X)
      RETURN
      END
```

```
C ERROR CALCULATES THE ERRORS OF SIGMA1,SIGMA2,PITCH, AND YAW.
C THIS ANALYSIS IS BASED ON TOTAL DIFFERENTIALS WITH
C QUADRATIC PERTURBATIONS NEGLECTED, AND WITH SMALL ANGLE APPROXIMATIONS
C SO THAT SIN(DELTA P)=DELTA P, AND COS(DELTA P)=1
      SUBROUTINE ERR(DT11,DT12,OMEGA)
      DOUBLE PRECISION X1,Y1,Z1,X2,Y2,Z2,XM,YM,ZM,SKEW1,
     1 SKEW2,BETA,RX1,RY1,RZ1,R1,RX2,RY2,RZ2,R2,COSMU,XR1,
     2 YR1,ZR1,XR2,YR2,ZR2,ANSWER,PI,RADIAN,
     3 ERROR,EWX,EWY,EWZ
      INTEGER OUTPUT
      COMMON X1,Y1,Z1,X2,Y2,Z2,XM,YM,ZM,SKEW1,
     1 SKEW2,BETA,RX1,RY1,RZ1,R1,RX2,RY2,RZ2,R2,COSMU,XR1,
     2 YR1,ZR1,XR2,YR2,ZR2,ANSWER(4,6),PI,RADIAN,
     3 ERROR,EWX,EWY,EWZ,NUMSOL,NREAD,OUTPUT
      DOUBLE PRECISION A,ALPHA,ARG,ATANG1,ATANG2,ATERM,A1,A2,BTERM,
     1 B1,B2,COSALF,COSG1,COSPIT,COSS1,COSS2,COSW,COSYAW,CP1,CP2,CTERM,
     2 CT1,CT2,C1,C2,DALF,DENOM,DG1,DG2,DMU,DPITCH,DP1,DP2,DSIG1,DSIG2,
     3 DTAN,DT1,DT11,DT12,DT2,DWDT12,DYAW,OMEGA,PHI1,PHI2,PITCH,RAD,
     4 SIGMA1,SINALF,SINMU,SINPIT,SINS1,SINS2,SINW,SINYAW,SP1,SP2,ST1,
     5 ST2,TANG1,TERM1,TERM2,TERM3,THETA1,THETA2,XNUMER,XYMAG,XYMAG1,
     6 XYMAG2,YAW
      DOUBLE PRECISION BOUND,DELP,DELY
      DOUBLE PRECISION S1OUT,S2OUT,DS1OUT,DS2OUT
      DO 1000 I=1,5
      BOUND=1.D0**(-I+1)
    9 FORMAT(" NOW INSIDE THE ERROR ANALYSIS ROUTINE, WITH ALL"/
     1 "  PERTURBATIONS SET AS "F10.5)
      WRITE(OUTPUT,9)BOUND
      DP1=BOUND
      DP2=BOUND
      DP2=BOUND
      DWDT12=BOUND
      DALF=BOUND
      DG1=BOUND
      DG2=BOUND
      DT1=BOUND
      DT2=BOUND
      THETA1=DATN2(XR1,YR1)
      THETA2=DATN2(XR2,YR2)
      XYMAG1=DSQRT(XR1**2+YR1**2)
      XYMAG2=DSQRT(XR2**2+YR2**2)
      PHI1=DATN2(ZR1,XYMAG1)
      PHI2=DATN2(ZR2,XYMAG2)
      CP1=DCOS(PHI1)
      CP2=DCOS(PHI2)
      SP1=DSIN(PHI1)
      SP2=DSIN(PHI2)
      ST1=DSIN(THETA1)
      ST2=DSIN(THETA2)
      CT1=DCOS(THETA1)
      CT2=DCOS(THETA2)
      YAW=DATN2(EWX,EWY)
      XYMAG=DSQRT(EWX**2+EWY**2)
      PITCH=DATN2(EWZ,XYMAG)
      COSPIT=DCOS(PITCH)
      SINPIT=DSIN(PITCH)
      COSYAW=DCOS(YAW)
      SINYAW=DSIN(YAW)
C CALCULATE MU AND DMU
       COSMU=XR1*XR2+YR1*YR2+ZR1*ZR2
       SINMU=DSQRT(1.D0-COSMU**2)
       XNUMER=SP1*ST1*(SP2*ST2*DP2-CP2*CT2*DT2)+
     1 CP2*ST2*(SP1*ST1*DP1-CP1*CT1*DT1)+
     2 CP1*CT1*(SP2*CT2*DP2+CP2*ST2*DT2)+
     3 CP2*CT2*(SP1*CT1*DP1+CP1*ST1*DT1)-
     4 CP1*SP2*DP1-SP1*CP2*DP2
       DMU=XNUMER/SINMU
```

```
C   CALCULATE DSIG1 THE ERROR OF SIGMA1
         ATANG1=DATAN(SKEW1)
         ATANG2=DATAN(SKEW2)
         ALPHA=OMEGA*DT11-BETA
         COSG1=DCOS(SKEW1)
         A=ATANG2/ATANG1
         COSALF=DCOS(ALPHA)
         SINALF=DSIN(ALPHA)
         TANG1=DTAN(SKEW1)
         RAD=DSQRT(1.D0+A*A+2.D0*A*COSALF)
         TERM1=TANG1*(COSALF*RAD/SINALF**2-A/RAD)*DALF
         TERM2=(RAD/(SINALF*COSG1*COSG1)-TANG1/(SINALF*RAD)*(A+COSALF)*
       1 ATANG2/ATANG1**2*(1.D0/(1.D0+SKEW1**2)))*DG1
         TERM3=TANG1/SINALF*(A+COSALF)/RAD/ATANG1/(1.D0+SKEW2**2)*DG2
         ARG=TANG1/SINALF*RAD
         DSIG1=1.D0/(1.D0+ARG**2)*(TERM1+TERM2+TERM3)
         SIGMA1=DATAN(ARG)
         S1OUT=SIGMA1*RADIAN
         DS1OUT=DSIG1*RADIAN
         WRITE(OUTPUT,10)SIGMA1,DSIG1,S1OUT,DS1OUT
      10 FORMAT("  SIGMA1="1PD23.15" RADIANS, WITH DSIG1="D23.15/
       1 "          ="0PF23.15" DEGREES            ="F23.15"DEGREES")
C    CALCULATE DSIG2 FROM IMPLICIT TOTAL DIFFERENTIAL OF THE
C    TWO STATION ASPECT ANGLE FORMULA
         COSS1=DCOS(SIGMA1)
         SINS1=DSIN(SIGMA1)
         COSS2=XM*XR1+YM*YR1+ZM*ZR1
         SINS2=DSQRT(1.D0-COSS2**2)
         COSW=DCOS(OMEGA*DT12)
         SINW=DSIN(OMEGA*DT12)
         DENOM=SINS1*COSS2*COSW-COSS1*SINS2
         ATERM=(SINS1*COSS2-COSS1*SINS2*COSW)*DSIG1
         BTERM=(SINS1*SINS2*SINW)*DWDT12
         CTERM=COSMU*DMU
         DSIG2=(ATERM+BTERM-CTERM)/DENOM
         SIGMA2=DASIN(SINS2)
         S2OUT=SIGMA2*RADIAN
         DS2OUT=DS2*RADIAN
         WRITE(OUTPUT,20)SIGMA2,DSIG2,S2OUT,DS2OUT
      20 FORMAT("  SIGMA2="1PD23.15" RADIANS, WITH DSIG2="D23.15/
       1 "          ="0PF23.15" DEGREES            ="F23.15" DEGREES")
         A1=-SINPIT*SINYAW*CP1*ST1-SINPIT*COSYAW*CT1+COSPIT*SP1
         B1=COSPIT*COSYAW*CP1*ST1-COSPIT*SINYAW*CP1*CT1
         C1=-SINS1*DSIG1+COSPIT*SINYAW*(SP1*ST1*DP1-CP1*CT1*DT1)
       1 +COSPIT*COSYAW*(SP1*CT1*DP1+CP1*ST1*DT1)-SINPIT*CP1*DP1
         A2=-SINPIT*SINYAW*CP2*ST2-SINPIT*COSYAW*CT2+COSPIT*SP2
         B2=COSPIT*COSYAW*CP2*ST2-COSPIT*SINYAW*CP2*CT2
         C2=-SINS2*DSIG2+COSPIT*SINYAW*(SP2*ST2*DP2-CP2*CT2*DT2)
       1 +COSPIT*COSYAW*(SP2*CT2*DP2+CP2*ST2*DT2)-SINPIT*CP2*DP2
         DENOM=A1*B2-A2*B1
         DPITCH=(C1*B2-C2*B1)/DENOM
         DYAW=(A1*C2-A2*C1)/DENOM
         DELP=DPITCH*RADIAN
         DELY=DYAW*RADIAN
         WRITE(OUTPUT,30)DPITCH,DYAW,DELP,DELY
      30 FORMAT("  THE ERROR OF PITCH IS"1PD23.15" YAW IS"D23.15" RADIANS"/
       1 "  IN DEGREES, THIS IS  "0PF23.15"       "F23.15)
    1000 CONTINUE
         RETURN
         END
         END$
```

APPENDIX II

LTPOS

Program to Determine Vehicle Position

(Interdata OS/32 Version)

```
$BATCH
C       PROGRAM LTPOS(3,85)
C
C       PROGRAM TO READ LASER TRACKER DATA TAPE, UNPACK DATA,
C       SMOOTH AND OUTPUT TO LINE PRINTER IN TABULAR FORM
C
C
        COMMON IH(32),IM(32),SEC(32),AZ(32),EL(32),RA(32),
       X        LIM(32),IROV(32),ISP(32),LSROW(32),ICMP(32),IAUTO(32),
       X        XL,YL,ZL,IH1,IM1,SEC1
C      X        PULST1(320),PULST2(320),EPOCH(320),PULSW(320),PULSP(320)
C
C
C       DOUBLE PRECISION IH,IM,SEC,IH1,IM1,SEC1
C      X                 EPOCH,PULSW,PULSP
C       INTEGER  PULST1,PULST2
        INTEGER*4 HR,WW,J,MN,MS
        INTEGER*4 IH,IM,IH1,IM1
        DIMENSION IP(5),IJ(2)
        DIMENSION IST(6)
        DIMENSION WW(448),ISEC(32),IMS(32)
        EQUIVALENCE (IP(1),LU),(IP(2),LP),(IP(3),NFILE)
        EQUIVALENCE (I,IJ(1)),(J,IJ(2)),(IP(4),NREC)
C
C
        DATA RADEG/57.29577951/,PI/3.141592654/,IPAGE/1/,IPNT/0/
        DATA PI2/6.2831853087/,AZ0/90./,EL0/0./
C
C       GET USER'S CONSOLE & LPR LU'S
C
        CALL CLOSE(8,IST1)
        CALL CLOSE(1,IST2)
        CALL CLOSE(6,IST3)
        CALL OPENW(1,'CON:',4,0,0,IST4)
        CALL OPENW(6,'PR:',2,0,0,IST5)
         CALL OPENW(8,'MAGO:',0,0,0,IST6)
        IF(IST1.NE.0) GO TO 400
125     CONTINUE
        IF(IST2.NE.0) GO TO 401
126     CONTINUE
        IF(IST3.NE.0) GO TO 402
127     CONTINUE
        IF(IST4.NE.0) GO TO 403
128     CONTINUE
        IF(IST5.NE.0) GO TO 404
129     CONTINUE
        IF(IST6.NE.0) GO TO 405
 130     CONTINUE
        LU=0
        LP=0
        NFILE=0
        NREC=0
        IF(LU.EQ.0) LU = 1
        IF(LP.EQ.0) LP = 6
        IF(NFILE.EQ.0) GOTO 2
        IF(NREC.EQ.0)  NREC  = 1
C
C       POSTSTION TAPE TO FILE & REC # (DEFAULTS = 1)
C
        CALL MTREL(NFILE,NREC)
C
```

```
C      READ FIRST TWO WORDS FOR T-0
C
    2 CONTINUE
C      CALL EXEC(1,110B,IJ,2)
       READ(8)WW
C 120  FORMAT(2A2)
C      SEPARATE INTO H,M,S
C      UNPACK DATA
       K=1
       L=K-1
       HR=0
       IARG1=14*L+1
       J=WW(IARG1)
       HR=HR+(ISHFT(J,-31))*20
       HR=HR+(ISHFT((ISHFT(J,1)),-31))*10
       HR=HR+(ISHFT((ISHFT(J,2)),-28))
       IH1 = HR
       MN=0
       MN=MN+(ISHFT(ISHFT(J,6),-31))*40
       MN=MN+(ISHFT(ISHFT(J,7),-31))*20
       MN=MN+(ISHFT(ISHFT(J,8),-31))*10
       MN=MN+(ISHFT(ISHFT(J,9),-31))*8
       MN=MN+(ISHFT(ISHFT(J,10),-31))*4
       MN=MN+(ISHFT(ISHFT(J,11),-31))*2
       MN=MN+(ISHFT(ISHFT(J,12),-31))*1
       IM1 =MN
       KSEC=0
       KSEC=KSEC+(ISHFT(ISHFT(J,13),-31))*40
       KSEC=KSEC+(ISHFT(ISHFT(J,14),-31))*20
       KSEC=KSEC+(ISHFT(ISHFT(J,15),-31))*10
       KSEC=KSEC+(ISHFT(ISHFT(J,16),-31))*8
       KSEC=KSEC+(ISHFT(ISHFT(J,17),-31))*4
       KSEC=KSEC+(ISHFT(ISHFT(J,18),-31))*2
       KSEC=KSEC+(ISHFT(ISHFT(J,19),-31))*1
   40  ISEC(K)=KSEC
       MS=0
       MS=MS+(ISHFT(ISHFT(J,20),-31))*800
       MS=MS+(ISHFT(ISHFT(J,21),-31))*400
       MS=MS+(ISHFT(ISHFT(J,22),-31))*200
       MS=MS+(ISHFT(ISHFT(J,23),-31))*100
       MS=MS+(ISHFT(ISHFT(J,24),-31))*80
       MS=MS+(ISHFT(ISHFT(J,25),-31))*40
       MS=MS+(ISHFT(ISHFT(J,26),-31))*20
       MS=MS+(ISHFT(ISHFT(J,27),-31))*10
       MS=MS+(ISHFT(ISHFT(J,28),-31))*8
       MS=MS+(ISHFT(ISHFT(J,29),-31))*4
       MS=MS+(ISHFT(ISHFT(J,30),-31))*2
       MS=MS+(ISHFT(ISHFT(J,31),-31))*1
       IMS(K)=MS
       SEC1=FLOAT(ISEC(K))+FLOAT(IMS(K))/1000.
C      IH1 = ISHFT(IAND(I,140000B),-14)*10 +
C     X          ISHFT(IAND(I,36000B),-10)
C      IM1    = ISHFT(IAND(I,1600B),-7)*10 +
C     X          ISHFT(IAND(I,170B),-3)
C      SEC1   = FLOAT(IAND(I,7B))*10. +
C     X          FLOAT(ISHFT(IAND(J,170000B),-12)) +
C     X          FLOAT(ISHFT(IAND(J,7400B),-8))/10. +
C     X          FLOAT(ISHFT(IAND(J,360B),-4))/100. +
C     X          FLOAT(IAND(J,17B))/1000.
C      WRITE(6,998)IH1,IM1,SEC1
C998   FORMAT(2X,I20,2X,I20,2X,F15.3)
       CALL HEADR(LU,LF,IH1,IM1,SEC1)
C      BACK UP TO START OF RECORD
```

```
C       CALL EXEC(3,2108)
        REWIND 8
C       WRITE(LU,100)
C 100   FORMAT('ENTER AZIMUTH,ELEVATION OFFSET (DEG)')
C       READ(LU,*)AZO,ELO
        AZO = AZO/RADEG
        ELO = ELO/RADEG
C       GET LAUNCHER COORDS.
C       WRITE(LU,110)
C 110   FORMAT('ENTER LAUNCHER COORDINAIES (X,Y,Z)')
C       READ(LU,*)XL,YL,ZL
C
C       READ DATA RECORD
C
    1   CALL UNPCK(IEOF)
C
C       OFFSET CORRECTIONS
C
        DO 10 I=1,32
C       AZIMUTH
        AZ(I) = (AZ(I)*360./262144.)/RADEG +AZO
        IF(AZ(I).GT.PI ) AZ(I)=AZ(I)-PI2
        IF(AZ(I).LT.-PI )  AZ(I)=AZ(I)+PI2
C       ELEVATION
        EL(I) = (EL(I)*360./262144.)/RADEG   +ELO
         IF(EL(I).GT.PI) EL(I)=EL(I)-PI2
        IF(EL(I).LT.-PI ) EL(I)=EL(I)+PI2
C       TILT
        AZ(I) = ATAN2 ( SIN(AZ(I))* COS(EL(I)) ,  COS(AZ(I))* COS(EL(I)))
        AZ(I) = AZ(I)*RADEG
        EL(I) = EL(I)*RADEG
C       WRITE(6,600)AZ(I),EL(I),RA(I)
C 600   FORMAT(10X,3(3X,F13.7))
   10   CONTINUE
        CALL OUTPT(LP,IEOF,IPNT,IPAGE)
        IF(IEOF.NE.1) GOTO 1
  999   REWIND 8
        WRITE(LP,200)
  200   FORMAT('1')
        GO TO 99
 400    WRITE(1,450) IST1
        GO TO 125
  450    FORMAT('DEVICE ASSIGNMENT ERROR***IST1=',I5)
  401    WRITE(1,452) IST2
         GO TO 126
  402    WRITE(1,453) IST3
         GO TO 127
  403    WRITE(1,454) IST4
         GO TO 128
  404    WRITE(1,455) IST5
         GO TO 129
  405    WRITE(1,456) IST6
         GO TO 130
  452    FORMAT('DEVICE ASSIGNMENT ERROR***IST2=',I5)
  453    FORMAT('DEVICE ASSIGNMENT ERROR***IST3=',I5)
  454    FORMAT('DEVICE ASSIGNMENT ERROR***IST4=',I5)
  455    FORMAT('DEVICE ASSIGNMENT ERROR***IST5=',I5)
  456    FORMAT('DEVICE ASSIGNMENT ERROR***IST6=',I5)
  99    CONTINUE
        END
        SUBROUTINE ITREL(NFILE,NREC)
        IFILE = NFILE - 1
        IREC  = NREC  - 1
```

```
C       FIND FILE
      1 CONTINUE
        IF(IFILE.LE.0) GOTO 10
C EXEC CALL REMOVED HERE
        IFILE = IFILE - 1
        GOTO 1
C       FIND RECORD #
     10 CONTINUE
        IF(IREC.LE.0) GOTO 99
C EXEC CALL REMOVED HERE
        IREC = IREC - 1
        GOTO 10
     99 RETURN
        END
         SUBROUTINE HEADR(LU,LP,IH,IM,SEC)
C
C       ROUTINE TO PRINT HEADER SHEET FOR LASER TRACKER
C       DATA LIST
C
        DIMENSION IPROJ(10),IDATE(5)
C
C       DOUBLE PRECISION IH,IM,SEC
        INTEGER*4 IH,IH1,IM,IM1
C
        WRITE(LU,100)
    100 FORMAT('ENTER PROJECT NAME')
        READ(LU,101)(IPROJ(K),K=1,10)
    101 FORMAT(10A2)
        WRITE(LU,110)
    110 FORMAT('ENTER TEST DATE (EX: 01 JAN 78)')
        READ(LU,111)(IDATE(K),K=1,5)
        WRITE(LU,120)
   120  FORMAT('TIME OF TEST')
        READ(LU,121)IH,IM,SEC
    121 FORMAT(2(I2,X),F6.3)
    111 FORMAT(5A2)
        WRITE(LP,200)
    200 FORMAT(1H1,20(/),
       X        40X,'US ARMY MISSLE RESEARCH AND DEVELOPMENT COMMAND',/,
       X         51X,'TECHNOLOGY LABORATORY',/,
       X        48X,'SYSTEMS SIMULATION DIRECTORATE',/,
       X       53X,'SYSTEMS EVALUATION',//,
       X        50X,'LAMPAMS - LASER TRACKER',//)
        WRITE(LP,210)(IPROJ(K),K=1,10),(IDATE(K),K=1,5),IH,IM,SEC
    210 FORMAT(15X,10A2,23X,5A2,26X,2(I2,':'),F6.3)
        RETURN
        END
        SUBROUTINE UNPCK(IEOF)
C
C       ROUTINE TO READ LAMPAM'S DATA TAPE AND UNPACK DATA
C       INTO VARIABLES (HOURS,MINUTES,SEC,AZ,EL,RA, ETC.)
C
C       RETURNS:
C         IEOF = 0 IF MORE DATA ON TAPE
C         IEOF = 1 IF END OF FILE DETECTED WITH READ
C
C
        COMMON IH(32),IM(32),SEC(32),AZ(32),EL(32),RA(32),
       X        LIM(32),IROV(32),ISP(32),LSRON(32),ICMP(32),IAUTO(32),
       X        XL,YL,ZL,IH1,IM1,SEC1
C      X        PULST1(320),PULST2(320),EPOCH(320),PULSW(320),PULSP(320)
C
```

```
      DOUBLE PRECISION IH,IM,SEC,IH1,IM1,SEC1
      DOUBLE PRECISION EPOCH,PULSW,PULSP

      DIMENSION WW(448),ISEC(32),IMS(32)
      INTEGER*4 HR,WW,J,MN,MS
      INTEGER*4 IH,IM,IH1,IM1
      INTEGER PULST1,PULST2


      IEOF = 0

      READ RECORD

      J = 0
1      READ(8)WW

      CHECK FOR EOF

      CALL EXEC(13,8,IEQT5)    ******
      IEQT5=IAND(IEQT5,200B)        ******
      IF(IEQT5.NE.0) IEOF = 1
      IF(IEOF.EQ.1) GOTO 99

      UNPACK DATA
      DO 125 K=1,32
      L=K-1
0     HR=0
      IARG1=14*L+1
      J=WW(IARG1)
      HR=HR+(ISHFT(J,-31))*20
      HR=HR+(ISHFT((ISHFT(J,1)),-31))*10
      HR=HR+(ISHFT((ISHFT(J,2)),-28))
       IH(K)=FLOAT(HR)-IH1
      MN=0
      MN=MN+(ISHFT(ISHFT(J,6),-31))*40
      MN=MN+(ISHFT(ISHFT(J,7),-31))*20
      MN=MN+(ISHFT(ISHFT(J,8),-31))*10
      MN=MN+(ISHFT(ISHFT(J,9),-31))*8
      MN=MN+(ISHFT(ISHFT(J,10),-31))*4
      MN=MN+(ISHFT(ISHFT(J,11),-31))*2
      MN=MN+(ISHFT(ISHFT(J,12),-31))*1
10     IM(K)=MN-IM1
      KSEC=0
      KSEC=KSEC+(ISHFT(ISHFT(J,13),-31))*40
      KSEC=KSEC+(ISHFT(ISHFT(J,14),-31))*20
      KSEC=KSEC+(ISHFT(ISHFT(J,15),-31))*10
      KSEC=KSEC+(ISHFT(ISHFT(J,16),-31))*8
      KSEC=KSEC+(ISHFT(ISHFT(J,17),-31))*4
      KSEC=KSEC+(ISHFT(ISHFT(J,18),-31))*2
      KSEC=KSEC+(ISHFT(ISHFT(J,19),-31))*1
0     ISEC(K)=KSEC
      MS=0
      MS=MS+(ISHFT(ISHFT(J,20),-31))*800
      MS=MS+(ISHFT(ISHFT(J,21),-31))*400
      MS=MS+(ISHFT(ISHFT(J,22),-31))*200
      MS=MS+(ISHFT(ISHFT(J,23),-31))*100
      MS=MS+(ISHFT(ISHFT(J,24),-31))*80
      MS=MS+(ISHFT(ISHFT(J,25),-31))*40
      MS=MS+(ISHFT(ISHFT(J,26),-31))*20
      MS=MS+(ISHFT(ISHFT(J,27),-31))*10
      MS=MS+(ISHFT(ISHFT(J,28),-31))*8
      MS=MS+(ISHFT(ISHFT(J,29),-31))*4
```

```
      MS=MS+(ISHFT(ISHFT(J,31),-31))*1
   50  IMS(K)=MS
      SEC(K)=FLOAT(ISEC(K))+FLOAT(IMS(K))/1000.-SEC1
      IWW1=ISHFT(ISHFT(ISHFT(WW(14*L+3),16),-30),16)
      IWW2=ISHFT(WW(14*L+2),-16)
   60 AZ(K)=FLOAT(IOR(IWW1,IWW2))
      IWW1=ISHFT(ISHFT(WW(14*L+2),16),-16)
      IWW2=ISHFT(ISHFT(ISHFT(WW(14*L+3),18),-30),16)
   70 EL(K)=FLOAT(IOR(IWW1,IWW2))
      IWW1=ISHFT(WW(14*L+3),-16)
      IWW2=ISHFT(ISHFT(ISHFT(WW(14*L+3),20),-30),16)
   80 RA(K)=FLOAT(IOR(IWW1,IWW2))
      LIM(K)=ISHFT(ISHFT(WW(14*L+3),23),-31)
C     IANS(K,9)=ISHFT(ISHFT(WW(14*L+3),24),-30)
C     IANS(K,10)=ISHFT(ISHFT(WW(14*L+3),26),-30)
      IROV(K)=ISHFT(ISHFT(WW(14*L+3),28),-31)
      ISP(K)=ISHFT(ISHFT(WW(14*L+3),29),-31)
C     IANS(K,13)=ISHFT(ISHFT(WW(14*L+3),30),-31)
C     IANS(K,14)=ISHFT(ISHFT(WW(14*L+3),31),-31)
C 90  IANS(K,15)=ISHFT(WW(14*L+4),-27)
C     IANS(K,16)=ISHFT(ISHFT(WW(14*L+4),5),-27)
C     IANS(K,17)=ISHFT(ISHFT(WW(14*L+4),10),-31)
C     IANS(K,18)=ISHFT(ISHFT(WW(14*L+4),11),-31)
      LSRON(K)=ISHFT(ISHFT(WW(14*L+4),12),-31)
  100 ICMP(K)=ISHFT(ISHFT(WW(14*L+4),13),-31)
      IAUTO(K)=ISHFT(ISHFT(WW(14*L+4),14),-31)
C     IANS(K,22)=ISHFT(ISHFT(WW(14*L+4),15),-31)
C     IANS(K,23)=ISHFT(ISHFT(WW(14*L+4),16),-24)
C     IANS(K,24)=ISHFT(ISHFT(WW(14*L+4),24),-29)
C 110   IANS(K,25)=ISHFT(ISHFT(WW(14*L+4),27),-27)
       M=0
      DO 125 I=1,46,5
      J=I-1
       M=M+1
      IC=M+4+14*L
C     IANS(K,J+26)=ISHFT(WW(IC),-31)
C     IANS(K,J+27)=ISHFT(ISHFT(WW(IC),1)-28)
C     IANS(K,J+28)=(ISHFT(ISHFT(WW(IC),5),-21))*50.0E-9
C     IANS(K,J+29)=ISHFT(ISHFT(WW(IC),16),-31)
C120  IANS(K,J+30)=(ISHFT(ISHFT(WW(IC),17),-16))*50.0E-9
 125  CONTINUE
C
C
C     WRITE(1,101)
C101  FORMAT(5X,'SUBROUTINE UNPCK OK')
   99 RETURN
      END
      SUBROUTINE OUTPT(LP,IEOF,IPNT,IPAGE)
C
C
      COMMON IH(32),IM(32),SEC(32),AZ(32),EL(32),RA(32),
     X       LIM(32),IROV(32),ISP(32),LSRON(32),ICMP(32),IAUTO(32),
     X       XL,YL,ZL,IH1,IM1,SEC1
C    X       PULST1(320),PULST2(320),EPOCH(320),PULSW(320),PULSP(320)
C
C     DOUBLE PRECISION IH,IM,SEC,IH1,IM1,SEC1
      INTEGER*4 IH,IM,IH1,IM1
C    X                  EPOCH,PULSW,PULSP
C     INTEGER PULST1,PULST2
C
      DIMENSION TIME(100),X(100),XD(100),XDD(100),Y(100),YD(100),
     X          YDD(100),Z(100),ZD(100),ZDD(100),JLIM(100),JROV(100),
     X          JSIG(100),JLSR(100),JCMP(100),JAUTO(100)
```

```
C       DIMENSION D(100)
C
C       DOUBLE PRECISION TIME
        REAL JK
C
C       LOAD OUTPUT BUFFERS
C       WRITE(1,105)
C105    FORMAT(5X,'MADE IT INTO SUBROUTINE OUTPT')
C
        DO 100 I=1,32
        IPNT = IPNT+1
        X(IPNT) = AZ(I)
        Y(IPNT) = EL(I)
        Z(IPNT) = RA(I)
        JLIM(IPNT)=LIM(I)
        JROV(IPNT)=IROV(I)
        JSIG(IPNT)=ISP(I)
        JLSR(IPNT)=LSRON(I)
        JCMP(IPNT)=ICMP(I)
        JAUTO(IPNT)=IAUTO(I)
        TIME(IPNT)=(IH(I))*3600. +(IM(I))*60.+SEC(I)
C       WRITE(6,500)(EPOCH(K),PULST1(K),PULST2(K),PULSW(K),
C     X                  PULSP(K),K=1,320)
C 500   FORMAT(X,F22.11,4(5X,I10),/,)
        IF(IPNT.LT.100) GOTO 100
      1 IF(IPNT.EQ.0) GOTO 99
C       WRITE(6,500)(XDD(K),YDD(K),ZDD(K),K=1,IPNT)
  500   FORMAT(3(5X,E13.7))
C       WRITE(1,106)
C106    FORMAT(2X,'MADE IT UP TO CALL TO FIT')
        CALL FIT(TIME,X,XD,XDD,Y,YD,YDD,Z,ZD,ZDD,IPNT)
C        WRITE(6,500)(XDD(K),YDD(K),ZDD(K),K=1,IPNT)
C       WRITE(1,501)
C 501   FORMAT(' AFTER 'FIT'',//)
C       WRITE(6,500)(X(K),Y(K),Z(K),K=1,IPNT)
        DO 20 J = 1,IPNT
        CALL XYZCV(X(J),XD(J),XDD(J),Y(J),YD(J),YDD(J),Z(J),ZD(J),ZDD(J))
C       WRITE(1,107)
C107    FORMAT(2X,'BACK FROM XYZCV')
   20   CONTINUE
        DO 35 J = 1,IPNT,50
        JK=0.
        XS=0.
        YS=0.
        ZS=0.
        XV=0.
        YV=0.
        ZV=0.
        XA=0.
        YA=0.
        ZA=0.
        IPAGE = IPAGE+1
        CALL LABEL(LP,IPAGE)
        KK=J+49
        IF((IPNT.LT.100).AND.(J.GT.50))KK=IPNT
        IF(IPNT.LT.50) KK=IPNT
        DO 30 K=J,KK
        JK=JK+1.
        XS=XS+X(K)
        YS=YS+Y(K)
        ZS=ZS+Z(K)
        XV=XV+XD(K)
        YV=YV+YD(K)
```

```
      ZV=ZV+ZD(K)
      XA=XA+XDD(K)
      YA=YA+YDD(K)
      ZA=ZA+ZDD(K)
      WRITE(LP,600) TIME(K),X(K),Y(K),Z(K),XD(K),YD(K),ZD(K),
     X              XDD(K),YDD(K),ZDD(K),JLIM(K),JROV(K),JSIG(K),
     X              JLSR(K),JCMP(K),JAUTO(K)
   30 CONTINUE
      XS=XS/JK
      YS=YS/JK
      ZS=ZS/JK
      XV=XV/JK
      YV=YV/JK
      ZV=ZV/JK
      XA=XA/JK
      YA=YA/JK
      ZA=ZA/JK
   35 WRITE(LP,601)XS,YS,ZS,XV,YV,ZV,XA,YA,ZA
  601 FORMAT(/,' MEANS:',4X,9(X,F10.2))
      IPNT=0
C*******
C     IEOF=1
C*******
      IF(IEOF.EQ.1) GOTO 99
  100 CONTINUE
  600 FORMAT(X,F7.3,3X,9(X,F10.2),6I3)
      IF(IEOF.EQ.0) GOTO 99
      IF(IPNT.NE.0) GOTO 1
   99 RETURN
      END
      SUBROUTINE FIT(TIME,X,XD,XDD,Y,YD,YDD,Z,ZD,ZDD,IPNT)
C
C     ROUTINE TO SMOOTH LASER TRACKER DATA
C
      DOUBLE PRECISION CX(10),CY(10),CZ(10)
      DIMENSION X(100),XD(100),XDD(100),Y(100),YD(100),YDD(100)
      DIMENSION Z(100),ZD(100),ZDD(100)
      DOUBLE PRECISION TIME(100)
C     WRITE(6,600)IPNT,(X(I),Y(I),Z(I),I=1,IPNT)
  600 FORMAT(3X,I5,/,100(X,3(E13.7,5X),/,))
      IPNT1=IPNT-6
      DO 100 I=1,IPNT1
      IF(I.GT.7) GOTO 10
      IF(I.GT.1) GOTO 100
      CALL CRVFT(TIME(I),X(I),3,9,CX)
      CALL CRVFT(TIME(I),Y(I),3,9,CY)
      CALL CRVFT(TIME(I),Z(I),3,9,CZ)
      DO 50 J=1,7
      CALL FUNCT(TIME(J),X(J),XD(J),XDD(J),CX)
      CALL FUNCT(TIME(J),Y(J),YD(J),YDD(J),CY)
      CALL FUNCT(TIME(J),Z(J),ZD(J),ZDD(J),CZ)
   50 CONTINUE
      GOTO 100
   10 CONTINUE
      CALL CRVFT(TIME(I-6),X(I-6),3,9,CX)
      CALL CRVFT(TIME(I-6),Y(I-6),3,9,CY)
      CALL CRVFT(TIME(I-6),Z(I-6),3,9,CZ)
      CALL FUNCT(TIME(I),X(I),XD(I),XDD(I),CX)
      CALL FUNCT(TIME(I),Y(I),YD(I),YDD(I),CY)
      CALL FUNCT(TIME(I),Z(I),ZD(I),ZDD(I),CZ)
  100 CONTINUE
      IPNT1=IPNT-5
      DO 110 I=IPNT1,IPNT
```

```
      CALL FUNCT(TIME(I),X(I),XD(I),XDD(I),CX)
      CALL FUNCT(TIME(I),Y(I),YD(I),YDD(I),CY)
      CALL FUNCT(TIME(I),Z(I),ZD(I),ZDD(I),CZ)
  110 CONTINUE
      RETURN
      END
      SUBROUTINE CRVFT(X,Y,M,N,C)
C
C     LEAST-SQUARES POLYNOMIAL CURVE FITTING ROUTINE
C     SOLVES FOR COEFFICIENTS C(I) GIVEN X-Y PAIRS
C     OF DATA POINTS FOR EQUATIONS OF THE FORM:
C        Y=C(1)+C(2)*X+C(3)*X**2+...+C(M+1)*X**M
C
C     C = COEFFICIENT ARRAY
C     X = INDEPENDANT VARIABLE
C     Y = DEPENDANT VARIABLE
C     M = ORDER OF POLYNOMIAL (10 MAX)
C     N = NO. X-Y PAIRS
C
C
      DIMENSION X(10),Y(10)
      DOUBLE PRECISION A(11,11),B(11),P(20),YY(9),C(10)
      DOUBLE PRECISION FACTOR,SUM,TEMP,X
C
C
C     WRITE(6,661)M,N
  661 FORMAT(X,I6,5X,I6)
      DO 1 I=1,9
C     WRITE(6,666)X(I),Y(I),C(I)
  666 FORMAT(X,3(E13.7,5X))
    1 YY(I)=Y(I)
      NN=M+1
      DO 5 I=1,NN
    5 C(I)=0.
C     COMPUTE 'P' ARRAY (POWERS X(I) )
      MX2=M*2
      DO 13 I=1,MX2
      P(I)=0.
      DO 13 J=1,N
   13 P(I)=P(I)+X(J)**I
C     DEVELOP CONSTANT TERMS OF NORMAL EQNS.
      L=M+1
      DO 30 I=1,L
      DO 30 J=1,L
      K=I+J-2
      IF(K)29,29,28
   28 A(I,J)=P(K)
      GOTO 30
   29 A(1,1)=N
   30 CONTINUE
      B(1)=0.
      DO 21 J=1,N
   21 B(1)=B(1)+YY(J)
      DO 22 I=2,L
      B(I) = 0.
      DO 22 J=1,N
   22 B(I)=B(I)+YY(J)*X(J)**(I-1)
C     PIVOTAL CONDENSATION
      NM1=L-1
      DO 300 K=1,NM1
      KP1=K+1
      MX2=K
      DO 400 I=KP1,L
```

```
      IF(DABS(A(I,K))-DABS(A(MX2,K))) 400,400,401
  401 MX2=I
  400 CONTINUE
      IF(MX2-K)500,500,405
  405 DO 410 J=K,L
      TEMP=A(K,J)
      A(K,J)=A(MX2,J)
  410 A(MX2,J)=TEMP
      TEMP=B(K)
      B(K)=B(MX2)
      B(MX2)=TEMP
C     ELIMINATION AND BACK SOLUTION
  500 DO 300 I=KP1,L
      FACTOR=A(I,K)/A(K,K)
      A(I,K)=0.
      DO 301 J=KP1,L
  301 A(I,J)=A(I,J)-FACTOR*A(K,J)
  300 B(I)=B(I)-FACTOR*B(K)
      C(L)=B(L)/A(L,L)
      I=NM1
  710 IP1=I+1
      SUM=0.
      DO 700 J=IP1,L
  700 SUM=SUM+A(I,J)*C(J)
      C(I)=(B(I)-SUM)/A(I,I)
      I=I-1
      IF(I) 800,800,710
  800 CONTINUE
      RETURN
      END
      SUBROUTINE FUNCT(TIME,X,XD,XDD,C)
      DOUBLE PRECISION C(4)
      DOUBLE PRECISION TIME
C     WRITE(6,100)(C(K),K=1,4)
  100 FORMAT(15X,4(E13.7,3X),'***')
C     X=C(1) +C(2)*TIME +C(3)*TIME**2 +C(4)*TIME**3
      X=((C(4)*TIME+C(3))*TIME+C(2))*TIME+C(1)
C     XD=C(2) +2.*C(3)*TIME +3.*C(4)*TIME**2
      XD=(3.*C(4)*TIME+2.*C(3))*TIME+C(2)
      XDD=2.*C(3) +6.*C(4)*TIME
      RETURN
      END
      SUBROUTINE XYZCV(X,XD,XDD,Y,YD,YDD,Z,ZD,ZDD)
C
C
C     ROUTINE TO CONVERT TO RANGE CO-ORDINATE SYSTEM
C
C
C     XLT,YLT,ZLT = TRACKER CO-ORDS.
C     XL,YL,ZL = LAUNCHER CO-ORDS.
C
C
      COMMON IH(32),IM(32),SEC(32),AZ(32),EL(32),RA(32),
     X       LTM(32),IROV(32),ISP(32),LSRDN(32),ICMP(32),IAUTO(32),
     X       XL,YL,ZL,IH1,IM1,SEC1
C
C     DOUBLE PRECISION IH,IM,SEC,IH1,IM1,SEC1
      INTEGER*4 IH,IM,IH1,IM1
C
C     TRACKER CO-ORDS.
C
      XLT = 1072.125
      YLT = -2359.947
```

```
      ZLT =  23.885
C***********************************************
C      TRY RANGE COORDS FOR LAUNCHER (XL,YL,ZL)
      XL=0.
      YL=0.
      ZL=0.
C***********************************************
C
C      COMPUTE POS. OF ROUND (X,Y,Z)
C
C      WRITE(6,600)X,Y,Z
C 600  FORMAT(3(3X,E14.7))
      X=X*.01745329
      Y= Y*.01754329
      XD=XD*.01754329
      YD= YD*.01754329
      XDD=XDD*.01754329
      YDD= YDD*.01754329
      XM=XLT-XL+Z* COS(Y)* COS(X)
      YM=YLT-YL+Z* COS(Y)* SIN(X)
      ZM=ZLT-ZL+Z* SIN(Y)
C**********START THEIRS*****************
C
C      DETERMINE VELOCITIES
C
      VR=ZD
      VAZ=Z*XD* COS(Y)
      VEL=Z*YD
C
C      VELO IN LAUNCHER-CENTERED SYS
C
      VX=VR* COS(X)* COS(Y)-VAZ* SIN(X)-VEL* COS(X)* SIN(Y)
      VY=VR* SIN(X)* COS(Y)+VAZ* COS(X)-VEL* SIN(X)* SIN(Y)
      VZ=VR* SIN(Y)+VEL* COS(Y)
C
C      COMPUTE ACCELS.
C
      AR=ZDD-Z*YD**2-Z*XD**2*( COS(Y))**2
      AAZ=(Z*XDD+2*XD*ZD)* COS(Y)-
     X     2.*Z*XD* SIN(Y)*YD
      AEL=Z*YDD+2.*YD*ZD+Z* SIN(Y)* COS(Y)*XD**2
C
C      CONVERT TO LAUNCHER CO-ORDS.
C
      AX=AR* COS(X)* COS(Y)-AAZ* SIN(X)-AEL* COS(X)* SIN(Y)
      AY=AR* SIN(X)* COS(Y)+AAZ* COS(X)-AEL* SIN(X)* SIN(Y)
      AZ1=AR* SIN(Y)+AEL* COS(Y)
      X=XM
      Y=YM
      Z=ZM
      XD=VX
      YD=VY
      ZD=VZ
      XDD=AX
      YDD=AY
      ZDD=AZ1
C*************END THEIRS, START MINE*******************
C      VR=ZD
C      VAZ=XD
C      VEL=YD
C      XD=VR*COS(Y)*COS(X)-Z*VEL*SIN(Y)*COS(X)-Z*VAZ*COS(Y)*SIN(X)
C      YD=VR*COS(Y)*SIN(X)-Z*VEL*SIN(Y)*SIN(X)+Z*VAZ*COS(Y)*COS(X)
C      ZD=VR*SIN(Y)+Z*VEL*COS(Y)
```

```
C       AR=ZDD
C       AAZ=XDD
C       AEL=YDD
C       XDD=(AR-Z*(VEL**2+VAZ**2))*COS(Y)*COS(X)
C      X      +(AR-2.*VR*VEL)*SIN(Y)*COS(X)
C      X      +(Z*AAZ-2.*VR*VAZ)*COS(Y)*SIN(X)
C      X      -2.*Z*VEL*VAZ*SIN(Y)*SIN(X)
C       YDD=(AR-Z*VAZ**2)*COS(Y)*SIN(X)
C      X      -(VR*VEL+2.*Z*VEL*VAZ)*SIN(Y)*COS(X)
C      X      +(2.*VR*VAZ-Z*(VEL**2-AAZ))*COS(Y)*COS(X)
C      X      -(VR*VEL+Z*AEL)*SIN(Y)*SIN(X)
C       ZDD=(AR-Z*VEL**2)*SIN(Y)+(2.*VR*VEL+Z*AEL)*COS(Y)
C       X=XM
C       Y=YM
C       Z=ZM
C*************END MINE*************************
C       WRITE(6,601)X,Y,Z
C 601   FORMAT(3(3X,E14.7),/)
        RETURN
        END
        SUBROUTINE LABEL(LP,IPAGE)
C
C
C       ROUTINE TO PRINT DATA PAGE LABEL & PG. #
C
C
        NPAGE = IPAGE -1
        WRITE(LP,100)NPAGE
  100    FORMAT(1H1,52X,'LAMPAMS - LASER TRACKER',19X,'PAGE: ',I4,//,
       X    3X,'TIME',17X,'POSITION',24X,'VELOCITY',23X,'ACCELERATION',
       X           15X,'STATUS',/,
       X    3X,'(SEC)',19X,'(FT)',26X,'(FT/SEC)',23X,'(FT/SEC-SEC)',/,
       X    17X,'X',10X,'Y',10X,'Z',10X,'X',10X,'Y',10X,'Z',10X,
       X           'X',10X,'Y',10X,'Z',5X,'LM OV SP ON CP AT',/,
       X    3X,'---- ',9(8X,'---'),3X,6(' --'),/)
        RETURN
        END
$BEND
```

US ARMY MISSLE RESEARCH AND DEVELOPMENT COMMAND
TECHNOLOGY LABORATORY
SYSTEMS SIMULATION DIRECTORATE
SYSTEMS EVALUATION

LAMPAMS - LASER TRACKER

TIME (SEC)	POSITION (FT) X	Y	Z	VELOCITY (FT/SEC) X	Y	Z
----	---	---	---	---	---	---
0.250	1067.57	259.27	25.45	58.11	-47.44	43.47
0.260	1068.51	258.56	26.08	36.92	-24.50	20.18
0.270	1069.06	258.26	26.30	19.13	-6.59	2.82
0.280	1069.29	258.26	26.23	4.72	6.28	-8.64
0.290	1069.27	258.47	25.99	-6.32	14.10	-14.19
0.300	1069.06	258.79	25.70	-13.99	16.87	-13.83
0.310	1068.74	259.12	25.48	-18.30	14.58	-7.57
0.320	1068.27	259.51	25.45	-6.47	5.34	5.12
0.330	1068.36	259.30	25.76	6.08	-9.46	14.32
0.340	1068.76	258.94	26.22	10.66	-7.66	15.49
0.350	1069.09	258.66	26.54	6.14	-4.14	7.96
0.360	1069.11	258.84	26.55	-2.11	-1.78	-3.58
0.370	1068.85	258.85	26.23	-7.17	1.68	-10.53
0.380	1068.53	258.97	25.86	-6.05	4.54	-9.86
0.390	1068.46	259.06	25.72	0.69	4.05	-3.85
0.400	1068.66	259.07	25.82	7.56	11.49	1.68
0.410	1068.98	259.37	26.00	9.91	-5.46	4.25
0.420	1069.22	259.23	26.11	5.00	4.20	2.54
0.430	1069.20	259.08	26.10	-4.05	-7.88	-3.04
0.440	1068.92	259.15	25.90	-11.38	2.78	-7.25
0.450	1068.55	258.95	25.66	-11.85	11.33	-7.16
0.460	1068.32	259.64	25.54	-5.03	2.64	-3.51
0.470	1068.37	259.59	25.57	4.51	-1.34	0.50
0.480	1068.66	259.34	25.67	12.40	-13.07	3.18

Note that target acceleration and LAMPAMS status report have not been presented but are normally included in the output.

```
1))*40
1))*20
1))*10
1))*8
31))*4
31))*2
31))*1

3),-31))*40
4),-31))*20
5),-31))*10
6),-31))*8
7),-31))*4
8),-31))*2
9),-31))*1

31))*800
31))*400
31))*200
31))*100
31))*80
31))*40
31))*20
31))*10
31))*8
31))*4
31))*2
```